A Mobile Agent Based Secure Aggregation in Sensor Network

NEERAJ KUMAR[1]
R.B. PATEL[2]
[1]School of Computer Science and Engineering, SMVD University, Katra (J&K), India
[2]Deptt. of Computer Science and Engineering, MM University, Mullana, Haryana, India
Email: [1]nehra04@yahoo.co.in, [2]patel_r_b@yahoo.com

Abstract. Compared with traditional wired and wireless networks, low-power wireless sensor networks (WSN) can be rapidly deployed in a large geographical area in a self-configured manner. In such types of networks data aggregation is a key issue. Hop-by-hop data aggregation in this regard is a very important technique for reducing the communication overhead and energy expenditure of sensor nodes during the process of data collection in a sensor network. However, because individual sensor readings are lost in the per-hop aggregation process, compromised nodes in the network may forge false values as the aggregation results of other nodes, resulting the base station into accepting wrong aggregation results. Here a fundamental challenge is how the base station can obtain a good approximation of the aggregated data when a group of sensor nodes are compromised. In this paper, we have designed and implemented a prototype for secure data aggregation using mobile agent (MA) technology. The key advantage of using MA is their capability to move in heterogeneous networks without consuming many resources. The designed scheme uses a novel probabilistic grouping technique to dynamically partition the nodes in a tree topology into multiple logical levels of similar sizes. A commitment-based hop-by-hop aggregation is performed by MA in each group to generate a group aggregate. Extensive analysis and simulations show that designed scheme can achieve the level of efficiency close to an ordinary hop-by-hop aggregation protocol while providing high assurance on the trustworthiness of the aggregation result. The prototype implementation on top of TinyOS shows that designed scheme is practical on current sensor nodes such as Mica2 motes

Keywords: Aggregation, Security, Mobile Agent, Sensor Nodes, Cluster head.

(Received June 08, 2009/Accepted September 16, 2009)

1. Introduction

Wireless sensor networks (WSNs) are envisioned to be economic solutions to many important applications, such as real-time traffic monitoring, military surveillance, and homeland security [1]. A sensor network may consist of hundreds or even thousands of low-cost sensors, each of which acts as an information source, sensing and collecting data from the environment for a given task. There may also exist one or more base stations (or data sinks) which subscribe to specific data streams by distributing interests or queries. The sensors in the network then push relevant data to a querying base station (BS). However, it is very inefficient for every sensor node to report their raw data because every data packet need traverse many hops to reach the BS, especially considering that sensor nodes are often constrained by scarce resources in energy, communication, computation, and memory. On the other hand, as in many cases sensor nodes in an area detect the common phenomena; there is high redundancy in their raw data. Thus, reporting raw data back to the BS is often unnecessary. One of the data aggregation approach is Hop-by-hop aggregation. Hop-by- hop aggregation, however, opens a new door to false data injection attacks because Sensor nodes are often deployed in open and unattended environments, so they are vulnerable to physical tampering due to the low manufacturing cost. An adversary can obtain the confidential information (e.g., cryptographic keys) from a compromised sensor and reprogram it with malicious code. The compromised node may then report an arbitrary false fusion result to its parent node in the tree hierarchy, causing the final aggregation result to far deviate from the true measurement. This attack becomes more damaging when multiple compromised nodes collude in injecting false data. To answer this challenge, we propose a Secure Hop-by-hop Data Aggregation using MA for sensor networks. In the designed scheme, during a normal hop-by-hop aggregation process in a

tree hierarchy, we need to place more trust on high-level nodes than low-level nodes, because the aggregated result calculated by a high-level node is from a larger number of sensor nodes. In other words, if a compromised node is closer to the root, the bogus aggregated data from it will have a larger impact on the final result computed by the *BS* . However, in reality none of these low-cost sensors should be more trustworthy than others. As such, designed scheme takes the approach of reducing the trust on high level nodes. By using a probabilistic grouping method, proposed scheme dynamically partitions the topology tree into multiple logical clusters of similar sizes. Since fewer nodes will be under a high-level node in a logical subtree, the potential security threat from a compromised high-level node is reduced. To preserve the efficiency of per-hop aggregation, proposed scheme performs hop-by-hop aggregation in each logical group and generates one aggregate from each cluster.

The motivations behind the proposed system are: it will take a long time to aggregate the data at various levels in the WSN. So to overcome this difficulty we have used MAs. MAs have capabilities to travel in heterogeneous domains without much delay. Moreover, MAs are lightweight processes and so they do not induce any new overhead on the network and also consumes fewer resources. So keeping in mind of all these reasons, MAs based data aggregation architecture is proposed.

The rest of the paper is organized as follows: Section 2 discusses the related work, Section 3 discusses the network model, Section 4 describes the proposed system architecture, Section 5 discusses the detailed overview of the scheme, Section 6 discusses simulation and results analysis, and finally Section 7 concludes the article.

2 Related Work

Many data aggregation protocols [2][3][4][5] [6][7][8] have been proposed with security in mind. Hu and Evans [9] proposed a secure hop-by-hop data aggregation scheme that works if one node is compromised. Du et al. [10] proposed a mechanism that allows the base station to check the aggregated values submitted by several designated aggregators, based on the endorsements provided by a certain number of witness nodes around the aggregators. Przydatek et al. [11] presented SIA, a Secure Information Aggregation scheme for sensor networks where a fraction of sensors may be compromised. In their model, the aggregator collects the authenticated raw data from all the sensors in the network. The aggregator then computes an aggregation result over the raw data together with a commitment to the data based on a Merkle-hash tree and then sends them to a trustable remote home server, which later challenges the aggregator to verify the aggregate. Later on, Chan et al. [12] proposed a secure hierarchical data aggregation scheme for sensor networks. Roy et al. [13] augmented the normal data aggregation framework such as synopsis diffusion [14] with a set of countermeasures against values falsified by compromised nodes. They consider a ring topology for aggregation whereas ours is an aggregation tree. He et al. [15] devised privacy-preserving data aggregation schemes in sensor network, which is also interesting. Recently Dilip et.al [16], and Neeraj et. al.[17] propose an efficient cluster head election algorithm in WSNs. Also Data aggregation schemes are presented in WSNs[18-20]

3. Network Model

Because the assumptions of topology and topography used in most previous approaches are violated in realistic settings, we propose a new grid type network model. Figure 1 shows a network model consists of nodes with different capabilities and missions. The sensing nodes are assumed to be very limited in terms of memory and processing capability and perform the task of data collection. These nodes are indicated as white circles in Figure 1.

Cluster head nodes have more memory, processing ability, and additional radios. These nodes are equipped with additional keys and take on the role of routers and gateways between networks. Such a model presents a number of new possibilities for sensor networks.

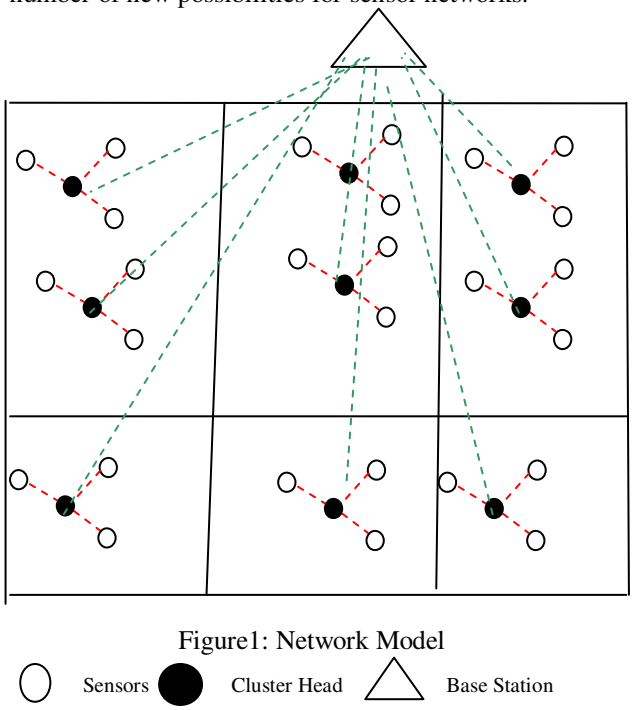

Figure1: Network Model

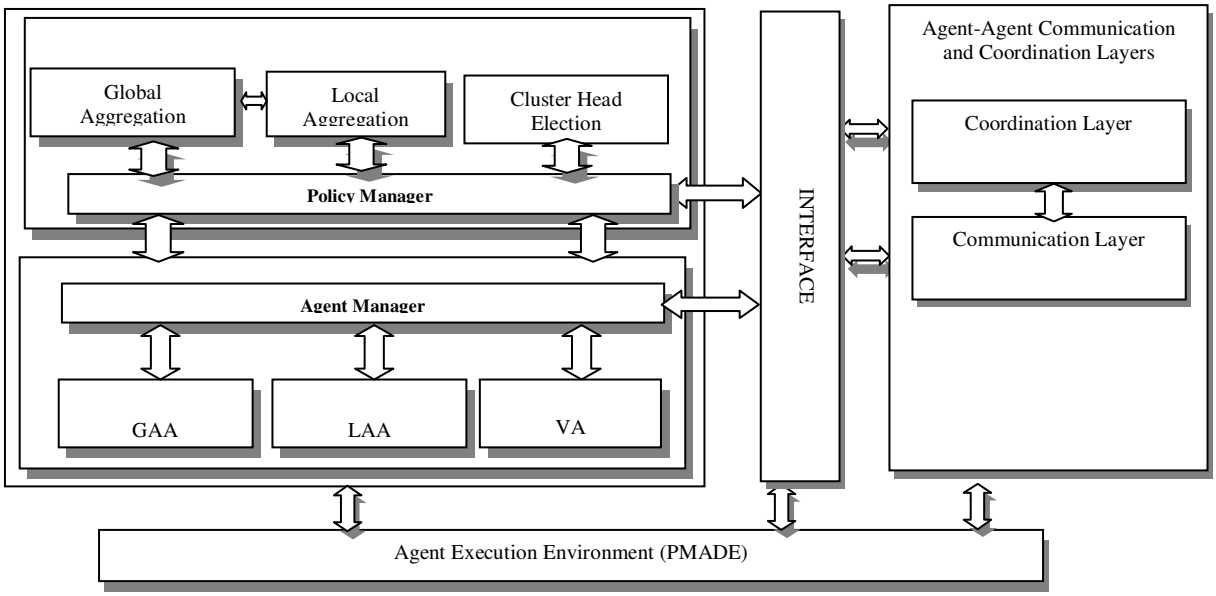

Figure2: Proposed System Architecture

Many of the techniques that have proven successful in ad hoc networks (e.g., secure routing [21], [22], [23], [24]) can now be incorporated into sensor networks.

4. Overview of the Proposed Approach

This section discusses the proposed system architecture {Figure 2} used for data aggregation using MA and its components. Figure 2 describes the agents and policies chosen for data aggregation along with the various layers and interface. The various components of the proposed architecture are presented and explained as follows in Figure 2:

Agent Manager- This is used to control the agents working in the proposed system. The various agents selected are- Global Aggregation Agent (GAA), Local Aggregation Agent (LAA) and Verification Agent (VA). Each agent executes the predefined policy defined in policy manager. There is a separate algorithm for data collection at respective level, i.e., base station (BS), cluster head (CH) and sensor nodes. GAA is used to aggregate the data between CH and BS while LAA is used to aggregate the data between CH and sensor nodes. Verification agent is used to verify the result collected by GAA and LAA. It operates on BS [Full description including algorithms for aggregation are explained in the coming section]

Policy Manager- This is used to control the policies associated with the agents. It controls the policy structure and frequency of policy to be executed by respective agent. The various policies selected in the proposed system are- Global aggregation, local aggregation and cluster head election. Each policy is executed by their respective agents who execute the corresponding algorithm at respective place.

Agent Execution environment- To control the movement of agents in sensor network, Platform for Mobile Agent Distribution and Execution (PMADE [25] is included in the architecture. PMADE provides various types of itinerary for MAs and their associated security and fault tolerance issues [25].

Agent-agent communication Layers- There are two layers included in the architecture for MAs for communication and coordination using mobile group approach [26]. This communication and coordination among agents is explained as follows:

Let P be the set of all possible MAs $\{MA_1, MA_2, MA_3, ..., MA_n\}$. These MAs are specific to the network selection and operates in groups for specific tasks. A mobile group is denoted by the set of agents $g = \sum_{i=1}^{k} MA_i$, $g \subset P$. The following operations are used by MAs for coordination and communication-

- join (g): issued by an agent, when it wants to join group *g*.
- l*eave (g)*: issued by an agent, when it wants to leave group *g*.
- move (g, l): issued when an agent wants to move from its current location-to-location l., where l is the location of agents.
- send (g, m): issued by an agent when it wants to multicast a message *m* to the members of group *g*.

- receive (g, m): issued by an agent to receive a message *m* multicast from the group *g*.

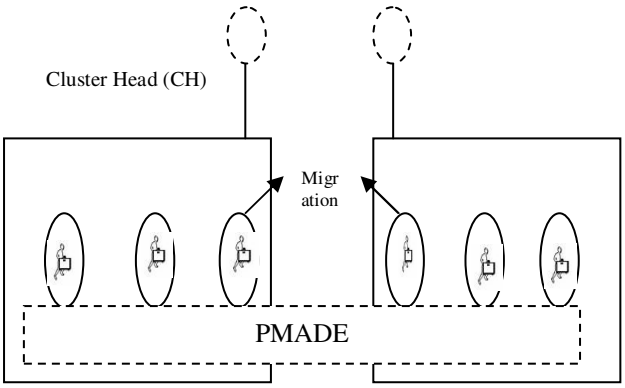

Figure3: Migration of MAs across Heterogeneous networks

As shown in Figure 3, MAs migrate in heterogeneous domains using PMADE [25]. PMADE provides the facility to create, deploy, migration, and itinerary of agents which is a part of proposed system architecture explained in Figure2. PMADE can be installed on BS to control, the mobility of agents. Agents are lightweight processes, so they did not consume any additional overhead on the network. Figure 3 is an abstract representation of Figure2, explaining only the mobility of agents.

5. Detailed Description of the Proposed Scheme

There are three levels for aggregation in the proposed scheme: at sensor node, at cluster head and at BS. The agents proposed in Figure 2 are actively involved in this phase to fulfill the task of aggregation. The all three levels are shown in Figure 4. The following steps are adopted for aggregation:

- Arrangement of nodes in hierarchical form
- Selection of Clusterhead at respective level
- Aggregation at leaf, intermediate and base station level
- Verification at BS

5.1 Arrangement of nodes into clusterhead in hierarchical form

Initially, the root node {Figure 4} broadcasts a tree construction message which includes its own id and its depth When a node, say i, receives a broadcast message first time from a node j, i assigns its depth to be the depth of j plus one, and its parent to be j. After this, it rebroadcasts the message. This process continues until all nodes have received this message. A count value is the aggregation of all nodes falling under a cluster head.

After constructing the tree, the BS transmits the query message through this tree. Besides the aggregation function that represents the BS request, a random number is added to the query. This random number is generated by the BS which is used for grouping as well as the query identification in the next phase. Specifically, a query packet that the BS broadcasts would be:

$BS \rightarrow *: F, S_g$ where F refers to a specific aggregation function, such as MEAN, SUM, and S_g is the random number generated for each query. The tree structure is constructed to reflect the data aggregation level by level i.e. from root to top. It also reflects the migration of agents at respective nodes.

5.2 Cluster Head Election procedure {Probabilistic Grouping}

In the previous phase, all nodes have identified their parents. In this phase, agents performs the Probabilistic grouping to conduct the selection of a cluster head for each group. This grouping is finished through the selection of cluster head nodes. Here we have a definition upon the cluster head node.

Definition: A cluster leader is the topmost node in a group, which completes and submits the aggregate result for the group. This node is changed among nodes and selected probabilistically during the process of data aggregation.

Cluster head are selected based on the count values and the grouping seed S_g received in the query dissemination phase. Two functions are used in selection. One is a cryptographically secure pseudorandom function P that uniformly maps the input values (node's id and S_g) into the range of [0, 1); the other is a grouping function P' that takes a positive integer (count) as the input and outputs a real number between [0, 1]. Each node, say i, decides if it is a cluster head by checking whether the following equality is true for it:

$$P(S_g \mid i) < P'(c) \quad \text{.....(1)}$$

If it is true, node i becomes a cluster head, and all the nodes in its sub tree that have not been grouped yet become members of its group.

A node with larger count has a higher probability to become a cluster head. The grouping function P' is

used to control the probability for a node to be chosen as a cluster head and it is preloaded in each sensor. In our construction, P' increases with the count value. Thus, if a node has a larger count value, the probability for it to become a cluster head is higher. The use of the random number S_g as the grouping seed is mainly for security and load balance. With the random number, the BS can change the cluster head among nodes instead of fixing their roles, so that the attacker cannot determine in advance which nodes will be the cluster head for each query. Otherwise, the attacker may target the cluster head and compromise them. Also, because a different S_g is used each time, every node is assigned into a different group that is formed on the fly. Another advantage of the proposed scheme is to balance the resource usage of nodes (e.g., storage, computation, and communication) to prolong the overall lifetime of the network.

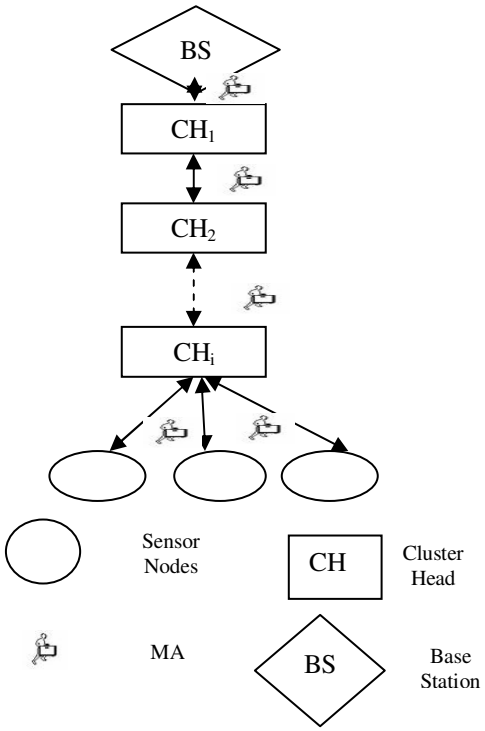

Figure4: Data Aggregation at various levels in Hierarchical Form

5.3 Aggregation at Leaf, Intermediate and Base Station

The aggregation at leaf and intermediate node is controlled by LAA. Each aggregation packet contains the sender's id, a data value, and a count value to indicate how many nodes contributing to the aggregated data. In addition, a one bit flag field in each packet is included to show whether the aggregate needs to be processed further by the nodes to the root. Flag value 1 means that no further aggregation is needed, whereas 0 means to be aggregated. This flag field is initialized to 0. After Cluster head finishes the aggregation for a group of nodes, this flag field is set to 1. Other nodes on the path to the root just forward those packets with flag 1. The pairwise key shared between each pair of parent and child is used to encrypt the aggregate data. Thus, using encryption saves the bandwidth that will otherwise be used for an additional MAC. In addition, a MAC is computed using the key shared with the BS which is attached at the end of each packet, which provides authentication to the BS.

Leaf Node Aggregation

In this case data aggregation is started by LAA from the leaf nodes in hierarchy towards the BS. Since a leaf node does not need to do aggregation, so LAA just sends its id, data, and count value to its parent (it also keeps a local copy of the packet). The packet that a LAA sends from node v_1 to its parent v_2 is as follows:

$v_1 \rightarrow v_2, 0, E(K_{v_1,v_2}, 1 | D_{v_1} | S_g) | MAC_{v_1}$ // format of the packet sent from node v_1 to its parent v_2

$MAC_{v_1} = MAC(K_{v_1}, 0 | 1 | v_1 | D_{v_1} | S_g$

where 0 is the aggregation flag, 1 is the count value(indication of participating nodes involved in aggregation), D_{v_1} is the reading of node v_1, K is the pariwise key shared between the participating nodes, and MAC_{v_1} is the MAC value computed by node v_1 with its individual key shared with the BS. Here S_g is included to identification of agent performing the task of aggregation. When the total aggregated value is reached at BS it will be quite easy to verify that which agent has sent which result and whether that agent is still working on that node or not.

Intermediate Node Aggregation

This step is also controlled by LAA. When an intermediate node receives an aggregation from its child node, it first checks the flag. If the flag is 0, it keeps a local copy of the aggregates and performs further aggregation; otherwise, the node directly forwards the packet to its parent node. It also performs checking on the validity of the count, D_{v_1} and S_g using the pairwise key shared between them.. If the aggregate packet does not pass this checking, it will discard the packet directly. Otherwise, it will further aggregate its own reading with all the aggregates carrying flag 0. The count value is updated as the sum of the count values in the received aggregates with flag 0 plus one using its own id and the new count as the inputs. The node then

encrypts the new count value and aggregation data using the pairwise key shared with its own parent. Let us assume that v_3 is the parent of v_2. The packet sent by LAA from v_2 to v_3 is :

$v_2 \rightarrow v_3 : v_2, 0, E(K_{v_2,v_3}, no\_of\_aggregated\_nodes \mid Agg_{v_1} \mid S_g) \mid MAC_{v_1}$

//format of the packet sent from v_2 to v_3

$Agg_{v1} = F(aggregate\_own\_othernodes)$

Where $aggregate\_own\_othernode$ is the aggregate value of its own node and other lower nodes in the hierarchy

$MAC_{v_1} = MAC(K_{v_1}, 0 \mid no\_of\_aggregated\_nodes \mid v_1 \mid Agg_{v_1} \mid MAC_i \oplus MAC_j \mid S_g)$

The MAC of an intermediate node is calculated over not only the previous fields but also the XOR of the MACs from its children ($MAC_i \oplus MAC_j$).

We have used a general aggregation function F instead of a specific one such as MEAN, SUM, or MEDIAN i.e. the designed scheme is applicable to multiple aggregation applications.

Base Station Aggregation

The aggregation process at this level is controlled by GAA. Like a regular intermediate node, GAA also computes a new aggregation, keeps the local copies of those packets with flag 0, and appends a corresponding MAC using its individual key for secure transmission. Unlike a regular intermediate node, it sets the flag to 1 in its aggregation packet and encrypts the new aggregate with its individual key shared with the BS and then cluster head node each level (say i) will send the packets to BS as follows:

$i \rightarrow BS : i, 1, E(K_i, count \mid Agg_i \mid S_g) \mid MAC_i$

$Agg_i = F(aggregate\_own\_othernodes)$

// format of the packet sent to BS

$MAC_i = MAC(K_i, 1 \mid count \mid i \mid Agg_i \mid MAC_i \oplus MAC_j \mid S_g)$

Where Agg_i is the aggregate result of the group and MAC_i is the MAC value computed by the cluster head node.

Based on the above aggregation rule, the aggregated data and the corresponding MACs are transmitted to the BS. There may be some nodes left without any group membership. In this case, the BS is the default head assigned by GAA for them. After the BS receives the aggregates from all groups, it decrypts and saves them in the following format: $(CH_i, C_i, Agg_i, MAC_i, S_g)$ where i is the cluster head node's id, C_i is the group count, Agg_i is the group aggregation value, MAC_i is the authentication tag computed by the cluster head.

5.4 Verification at Base Station

Verification at BS is done by VA. After the BS has received the aggregation messages from $CH_i, i = 1,2,...,n$ at each level VA verifies the authenticity of the aggregated value in each aggregation message. This includes verifying the content of the packet and the authenticity of $CH_i, i = 1,2,...,n$. First, based on the cluster head id, say i, in the message, the VA finds out the individual key of the node K_i from which it decrypts the data and gets the information $(i, C_i, Agg_i, MAC_i, S_g)$. The authenticity of the message is checked by the pariwise key shared between respective nodes Second, the VA also verifies the truth ness of the claimed $CH_i, i = 1,2,...,n$ by checking whether $P(S_g \mid i) < P'(c)$ because the BS knows P, and the grouping seed S_g. If this does not hold or any item in the packet is invalid, VA this information is passed to BS which simply drops the packet.

6. Simulation and Result Analysis

For simulation purpose, we have deployed N sensor nodes uniformly at random within 300 × 300 m target field, with $N = 200$ and n denotes the degree of polynomial stored in sensor node. Each sensor node has a constant transmission range of 20m, so that the degree of each node is approximately 15 ($N = 200$) on an average. We position a base station and a source node in opposite corners of the field, at a fixed point $(x, y) = (50,50)$. It is located approximately 15 hops away from each other. We distribute compromised nodes over 100m each side. Thus, compromised nodes are placed in between the base station and the source node. To evaluate the performance of proposed scheme, we run simulations of the proposed scheme on ns-2 [27]. We have used the typical TinyOS [28] with a little modification as a base routing protocol in the simulations. Each simulation experiment is conducted using different network topologies, and each result is averaged over 20 runs of different network topologies

6.1 Results Analysis

Overhead analysis of the Proposed Scheme

Computational Cost

During the aggregation, each node in the aggregation tree needs to compute one decryption, one pseudorandom function value, one grouping function value, one aggregation, one MAC, and one encryption. The encryption/decryption can directly use RC5 algorithm. The message authentication code (MAC) and pseudorandom function could be implemented by cipher block chaining CBC-MAC based on RC5. Furthermore, computation time spent on encryption and MAC are almost the same [29]. Therefore, during the aggregation, each node need compute several MACs (the computation of aggregation value and grouping function value only involves simple mathematical operations, so the cost is much less). The energy a sensor node uses in computing one MAC is about the same as that used for transmitting one byte [30]. Thus, from energy point of view, the energy used by a sensor node for aggregation is about the same as that used in transmitting MAC, so we believe this is a reasonable overhead for the current-generation sensor nodes.

Storage Overhead

Each node within groups keep local copies of packets with flag 0 received from children, except the leaf nodes which only keep a local copy of their own packets. Each aggregation packet is 24-byte (2 bytes for id, 9 bytes for data including count and grouping seed, 8 bytes for MACs, 5-byte for MA). Also, each node on the way to the BS need to construct a table recording the forwarding path, with each item having 13 bytes (2 bytes for cluster head node id, 2 bytes for incoming node's id, and 4 bytes for grouping seed, 5 byte for MA). Therefore, the total storage requirement for one node is at most several kilobytes.

Communication Overhead (Number of bytes transferred)

In this section, we focus on analyzing the communication overhead of the proposed scheme. Specifically, we first analyze the communication overhead of proposed scheme and then further use simulations to verify our claim that our scheme only causes little extra overhead compared to hop-by hop aggregation.

To accurately measure the overhead, we use the metrics of $packet * hop$ and $byte * hop$ (product of the data size and the message traveling distance), because message overhead is proportional to the traveling distance of sensing data. For ease of exposition, we do not consider the impact of packet retransmission due to the unreliable channel.

In the hop-by-hop data aggregation approach, the number of packets is equal to the number of edges in the broadcast tree. Hence, the communication overhead of the hop-by-hop aggregation approach is $O(n)$. The communication overhead of proposed scheme depends on the average group size g with in a cluster. If g is as large as n, the overhead is about $O(n)$. Otherwise, if g is small and can be treated as a constant number, the overhead is $O(n \log n)$. In either case, the overhead of proposed scheme is lower than the no-aggregation approach and slightly higher than the hop-by-hop aggregation approach as shown in Figure 5 and 6. As shown in Figures5, as the number of agents and number of cluster heads increases, the communication overhead (number of bytes transferred) also increases. But the overhead generated is still less than the case of no aggregation and comparable to hop by hop aggregation as shown in Figure 6. For the sake of simplicity, we have assigned one MA for each node in WSN. These agents communicate with other agents in WSN to accomplish their tasks using communication and coordination layers as proposed in Figure 2 of system architecture.

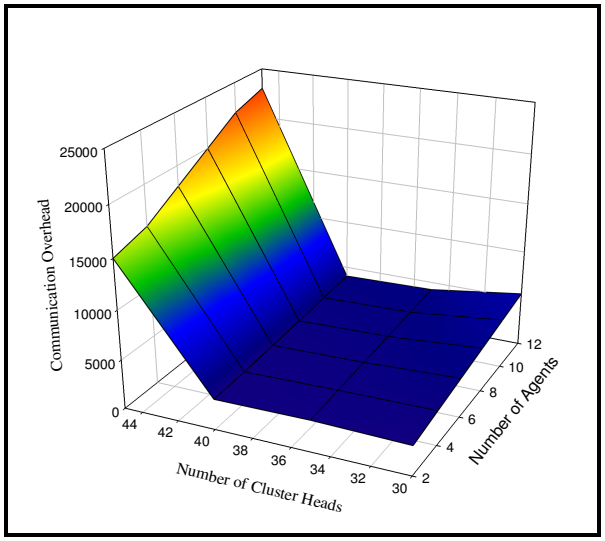

Figure5: Communication Overhead in proposed scheme with varying number of agents and number of Cluster heads

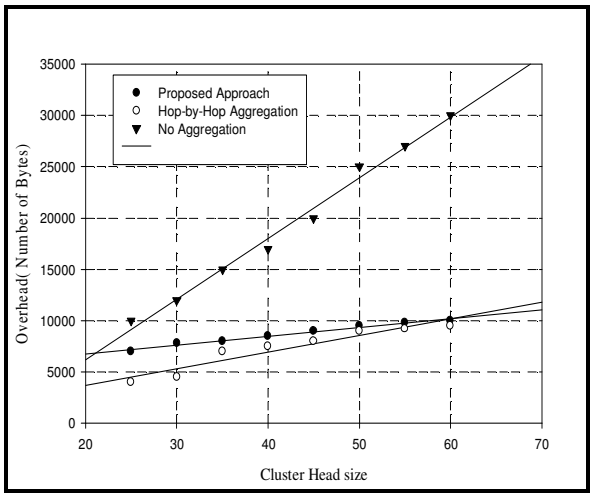

Figure6: Impact of increasing cluster head size on Overhead generated in the proposed approach

Detection Probability (%)

Figure 7 shows the detection probability of proposed scheme under multiple attacks. It is the ratio of number of malicious nodes detected to the total number of nodes available in WSN. Detection of malicious behavior is detected using the change in value of count. As soon as there is large change in the value of count, it reflects that there is certainly a malicious behavior of certain nodes exist in WSN. As it can be seen from Figure 7, the detection probability becomes higher with larger count changes (aggregation nodes), but this probability decreases with more malicious nodes. The reason is that with an increase in the number of malicious nodes the variance of group sizes with in a cluster raises, which causes a lower detection probability. When the number of malicious nodes is increased to more than half of the total number of groups, this probability is decreased to about 10%.

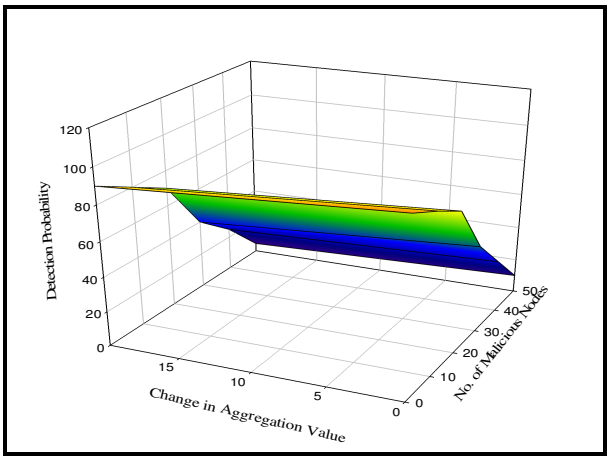

Figure7: Impact of change in count and malicious nodes on detection probability

Accuracy Improving Rate (In terms of number of detection)

It is defined as the rate at which detection of malicious behavior is noticed. It is measured as the number of malicious nodes detected in a finite interval of time. The system should be well capable of detection of such behavior and this detection rate should be continuously improving all the time.

From Figure 8 below, we can see that the accuracy improving rate increases with a larger aggregate change value, but decreases if there are more malicious nodes launching count changing attacks. As shown in Figure 8, the accuracy improving rate is higher with a larger count change value, but it is not influenced much by the increase in the number of malicious nodes.

Impact of Number of agents on Agent Trip Time

Agent trip time is a critical factor in networks such as WSN. It is total time taken by MA during its complete itinerary. Agent should complete its tasks in finite interval and then return these results to their launching hosts. So these metric is measured to see the effectiveness of the proposed scheme. So as the number of nodes to be visited and detection of malicious nodes increases, trip time also increases. Moreover, as shown in Figure 9, with the increase in number of agents and number of cluster heads, the agent trip time also increases. Because it will take a long time for agents to travel and pass the gathered information to BS for verification. Agents are light weight processes so they will not consume extra resources in terms of memory and processor in addition to the bandwidth usage in resource constrained WSN. Also the itinerary through which MAs travel is also secured using different has algorithms [6].

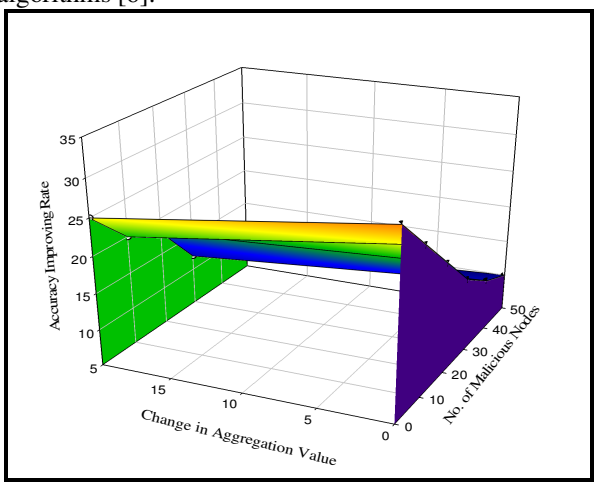

Figure8: Impact of change in aggregation value and no. of malicious nodes on accuracy improvement rate

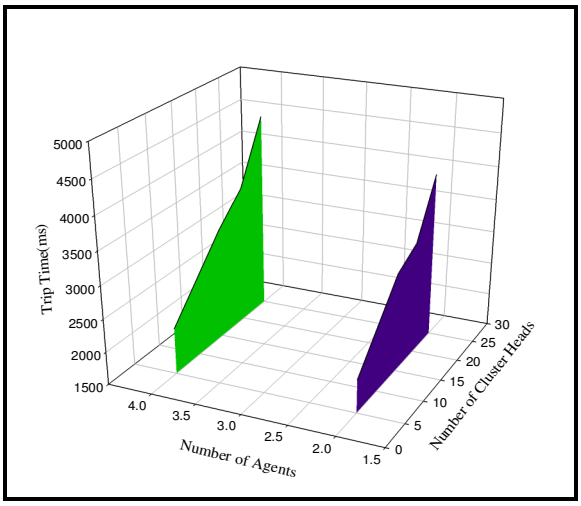

Figure9: Impact of Number of agents and Number of Cluster Heads on agent Trip time

Hence overall the storage and communication overhead will be reduced by using the proposed approach as seen in the above Figures 5and 6.

7. Conclusions

In this paper, we have proposed a Secure Data Aggregation Scheme for sensor networks using MA. The nodes are arranged in aggregation tree as BS, Cluster head nodes and sensor nodes. We partition the aggregation tree into cluster heads to reduce the communication and storage overhead. The data is aggregated at various levels by MAs. The aggregated data is verified at BS by the respective agent. Simulation results show that proposed scheme is effective in defending against count value changing attacks and is quite efficient with respect to the communication overhead generate and storage overhead. Also the use of MA does not slow down the overall aggregation time at respective levels. Hence the proposed system can be deployed at various sensor related applications.

References

[1] AKYILDIZ,I.,SU,W., ANKARASUBRAMANIAM, Y., AND CAYIRCI, E. Wireless sensor networks: A survey. Comput. Networks. 38: 4, 2002.

[2] ESTRIN, D., GOVINDAN, R., HEIDEMANN, J., AND KUMAR, S. Next century challenges: Scalable coordination in sensor networks. In Proceedings of ACM Mobicom (Mobicom'99) ACM, Seattle, Washington, 263–270, 1999.

[3] INTANAGONWIWAT, C., ESTRIN, D., GOVINDAN, R., AND HEIDEMANN, J. Impact of network density on data aggregation in wireless sensor networks. In proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS'02), 457–458, 2002.

[4] INTANAGONWIWAT, C., GOVINDAN, R., AND ESTRIN, D. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In proceedings of the International Conference on Mobile Computing and Networking (MobiCom'02), 56–67, 2000.

[5] KRISHNAMACHARI, B., ESTRIN, D., AND WICKER, S. The impact of data aggregation in wireless sensor networks. In proceedings of the International Workshop on Distributed Event- Based Systems, (DEBS'02). Vienna, Austria, 2002.

[6] MADDEN, S., FRANKLIN, M. J., HELLERSTEIN, J. M., AND HONG, W. TAG: A tiny aggregation service for ad-hoc sensor networks. In proceedings of the USENIX Symposium on Operating Systems Design and Implementation (OSDI'02), 2002.

[7] CHEN, J.-Y., PANDURANGAN, G., XU, D. Robust computation of aggregates in wireless sensor networks: distributed randomized algorithms and analysis. In proceedings of the International Symposium on Information Processing in Sensor Networks (IPSN'05), 348–355, 2005.

[8] YAO, Y. AND GEHRKE, J. The Cougar approach to in-network query processing in sensor networks. SIGMOD Record 31:3, 9–18, 2002.

[9] HU, L. AND EVANS, D. Secure aggregation for wireless networks. In proceedings of the Workshop on Security and Assurance in Ad Hoc Networks (SASN'03), 2003.

[10] DU, W., DENG, J., HAN, Y. S., AND VARSHNEY, P. K. A witness-based approach for data fusion assurance in wireless sensor networks. In proceedings of the Global Telecommunications Conference (GLOBECOM'03), 2003.

[11] PRZYDATEK, B., SONG, D., AND PERRIG, A. SIA: secure information aggregation in sensor networks. In proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys'03), 255–265, 2003.

[12] CHAN, H., PERRIG, A., AND SONG, D. Secure hierarchical in-network aggregation in sensor networks. In proceedings of the 13th ACM Conference on Computer and Communications Security (CCS'06), 278–287, 2006.

[13] ROY, S., SETIA, S., AND JAJODIA, S. Attack-resilient hierarchical data aggregation in sensor networks. In proceedings of the 4th ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'06), 71–82, 2006.

[14] NATH, S., GIBBONS, P., SESHAN, S., AND ANDERSON, Z. Synopsis diffusion for robust aggregation in sensor networks. In proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys'04), 250–262, 2004.

[15] HE, W., LIU, X., NGUYEN, H., NAHRSTEDT, K., AND ABDELZAHER, T. PDA: Privacy preserving data aggregation in wireless sensor networks. In proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'07), 2007.

[16] Dilip Kumar, T.C.Aseri, R.B. Patel. EECHE: Energy-Efficient Cluster Head Election Protocol for Heterogeneous Wireless Sensor Networks. In proceeding of International Conference on Advances in Computing, Communication and Control (ICAC3'09), Mumbai, India, 75-80, 2009.

[17] Neeraj Kumar, Manoj Kumar, R.B.Patel. Neural Network based energy efficient clustering and routing in wireless sensor networks. To appear in 1st IEEE international conference on network and communications (NetCom 09), 27-29 Dec. 2009, Chennai, India.

[18] Wen-Hwa Liao, Yucheng Kao, Chien-Ming Fan. Data aggregation in wireless sensor networks using ant colony algorithm. Journal of Network and Computer Applications, 31, 387–401, 2008.

[19] Chuan-Ming Liua,, Chuan-Hsiu Leea, Li-ChunWangb. Distributed clustering algorithms for data-gathering in wireless mobile sensor networks. Journal of Parallel and Distributed Computing, 67: 1187 – 1200, 2007.

[20] Peter Korteweg, Alberto Marchetti-Spaccamela, Leen Stougie, Andrea Vitaletti. Data Aggregation in Sensor Networks: Balancing Communication and Delay Costs. Theoretical Computer Science, 2008.

[21] S. Capkun and J. Hubaux. BISS: Building Secure Routing Out of an Incomplete Set of Security Associations. Proc. ACM Workshop Wireless Security (WiSe '03), Sept. 2003

[22] Y. Hu, A. Perrig, and D. Johnson. Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks. In proceeding of the MobiCom, 2002.

[23] P. Papadimitratos and Z. Haas. Secure Routing for Mobile Ad Hoc Networks. In proceeding of the SCS Comm. Networks and Distributed Systems Modeling and Simulation Conference, 2002.

[24] K. Sanzgiri, B. Dahill, B. Levine, C. Shields, and E. Belding-Royer. A Secure Routing Protocol for Ad Hoc Networks. In proceeding of the IEEE Int'l Conf. Network Protocols (ICNP '02), 2002.

[25] R.B. Patel and K. Garg, PMADE – A Platform for mobile agent Distribution & Execution. In Proceedings of 5th World Multi Conference on Systemics, Cybernetics and Informatics (SCI2001) and 7th International Conference on Information System Analysis and Synthesis (ISAS 2001),Orlando, Florida, USA, July 22-25, 4: 287-293, 2001.

[26] Raimundo, J. et al. The mobile groups approach for the coordination of mobile agents. Journal of Parallel and Distributed Computing, 65:1, 275-288, 2005.

[27] K. Fall and K. Varadhan (editors). NS notes and documentation. The VINT project, LBL, Feb 2000, http://www.isi.edu/nsnam/ns/ .

[28] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System Architecture Directions for Networked Sensors. ACU ASPLOS IX, November 2000.

[29] PERRIG, A., Szewczyk, R., WEN, V., CULLER, D., AND TYGAR, J. D. SPINS: Security protocols for sensor networks. In proceedings of the International Conference on Mobile Computing and Networking (MobiCom'01), 2001.

[30] YE, F., LUO, H., LU, S., AND ZHANG, L. Statistical en-route filtering of injected false data in sensor networks. In proceedings of the Annual Joint Conference of IEEE Computer and Communications Societies (INFOCOM'04), 2004.