SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks (2010)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.277

SPECIAL ISSUE PAPER

An efficient approach of secure group association management in densely deployed heterogeneous distributed sensor network

Al-Sakib Khan Pathan[1*,†], Muhammad Mostafa Monowar[2], Jinfang Jiang[3], Lei Shu[4] and Guangjie Han[3]

[1] Department of Computer Science, International Islamic University, Malaysia
[2] Department of Computer Engineering, Kyung Hee University, South Korea
[3] Department of Information & Communication Systems, Hohai University, Changzhou, China
[4] Graduate School of Information Science and Technology, Osaka University, Japan

ABSTRACT

A heterogeneous distributed sensor network (HDSN) is a type of distributed sensor network where sensors with different deployment groups and different functional types participate at the same time. In other words, the sensors are divided into different deployment groups according to different types of data transmissions, but they cooperate with each other within and out of their respective groups. However, in traditional heterogeneous sensor networks, the classification is based on transmission range, energy level, computation ability, and sensing range. Taking this model into account, we propose a secure group association authentication mechanism using one-way accumulator which ensures that: before collaborating for a particular task, any pair of nodes in the same deployment group can verify the legitimacy of group association of each other. Secure addition and deletion of sensors are also supported in this approach. In addition, a policy-based sensor addition procedure is also suggested. For secure handling of disconnected nodes of a group, we use an efficient pairwise key derivation scheme to resist any adversary's attempt. Along with proposing our mechanism, we also discuss the characteristics of HDSN, its scopes, applicability, future, and challenges. The efficiency of our security management approach is also demonstrated with performance evaluation and analysis. Copyright © 2010 John Wiley & Sons, Ltd.

KEYWORDS

mobile applications; pervasive computing; wireless sensor networks; security and privacy protection; verification

***Correspondence**

Al-Sakib Khan Pathan, Computer Science Department, International Islamic University Malaysia (IIUM), Jalan Gombak 53100, Kuala Lumpur, Malaysia.

E-mails: spathan@ieee.org,sakib@iium.edu.my

[†]Assistant Professor.

1. INTRODUCTION

Wireless sensor networks (WSNs) are composed of hundreds or thousands of inexpensive, low-powered sensing devices with limited computational capacity and communication bandwidth. Typical tasks of sensor nodes are sensing certain parameters from their surrounding environments and sending the readings to a central entity called base station or sink. With the rapid advancements of wireless technologies and sophistication of sensing technologies, the requirements of information types, data accuracy, and security have also been increased rapidly. Though some applications focus on collecting a specific type of data, utilization of various types of data could be more beneficial for extracting accurate and timely information. For example, in a volcano monitoring application, only one type of data, such as temperature, may be satisfactory for monitoring. But the average temperature of a certain region along with the seismic and acoustic readings can provide more comprehensive information regarding an imminent event. Again, for most of the applications, such as target tracking, environmental monitoring, and patient monitoring in hospitals, it is necessary to acquire different types of data from the same geographical region. Hence, multiple high-precision and security information need to be provided by WSN in many applications and network settings.

In order to get multiple types of data, ExScal mote [1,2] is designed by CrossBow Inc. and Ohio State University. This

mote is basically an extension of the well-known MICA2 mote [3] which supports multiple sensors (i.e., sensing units) on the same radio board. However, with the capabilities of today's sensors, instead of using this type of multipurpose node in the network, using different types of nodes that could collect data independently in the same area could be more efficient considering the utilization of memory, processing, and energy resources of the network. Now, we have various sensors that can monitor temperature, pressure, humidity, soil makeup, vehicular movement, noise level, lighting condition, the presence or absence of certain kinds of objects, mechanical stress level on attached objects, and other properties. Many innovative applications could benefit a lot using the smart sensors. For example, in the application of wind tunnel monitoring, sensor nodes are responsible to collect temperature, humidity, light, pressure, and other environmental parameters. One single kind of node usually cannot complete the measurement and recognition task of the environmental objectives, and only partial or one-sided information can be received; hence the amount of information that is received is very limited. In addition, each WSN node is also subject to quality, performance, and noise of the wind tunnel, so the information collected is often faced with greater uncertainty or even wrong. In such a situation, using a certain number of WSN nodes collecting different information separately could avoid the above problems. We have provided more points in the next section to support this argument.

The key point here is that whatever the configurations of the sensors are, data security is often deemed to be the most important aspect. Because sensor networks often constitute an information source that is a mission critical system component and thus, require commensurate security protection. If an adversary can thwart the work of the network by perturbing the generated information, stopping production, or pilfering information, then the usefulness of sensor networks is drastically curtailed. In order to guarantee data security or network security, many mechanisms have been proposed, such as intrusion detection system (IDS), password authentication, and so on. But first, it should be made sure that the sensors that are participating in the data acquisition and supplying process are authentic and are included as legitimate entities in the network. To be specific, along with other supporting security mechanisms, it is required to verify the authenticity of the sensors before allowing them to partake in any collaborative task. In this paper, we propose a secure group association authentication mechanism to guarantee the security of sensor nodes.

The remainder of this paper is structured as follows: Section 2 presents the related works; heterogeneous distributed sensor network (HDSN) is explained in Section 3; network assumptions and preliminaries are presented in Sections 4 and 5; Section 6 presents our approach of secure group association management in HDSN; performance analysis and discussions are presented in Section 7, and finally, Section 8 concludes the paper delineating the achievements from this work with future research directions on various facets of HDSN.

2. RELATED WORKS

Security is an important issue in WSNs. To ensure security in a WSN, it is essential to encrypt messages and authenticate the communicating nodes. For example, in Ref. [4], the authors propose a Random Pairwise Key scheme, which perfectly preserves the secrecy of the rest of the network when any node is captured. Later on, a closest pairwise-key predistribution scheme and a location-based pairwise-key scheme [5] came as the alternative to Random Pairwise Key scheme. This scheme takes advantage of the location information to improve the key connectivity. Also, another algorithm is presented that exploits location information to implement key management. In Ref. [6], a group-based key management model takes advantage of hexagonal grid and expected location information not only to reduce the memory cost but also to get better resilience against captured node attack. However, it has some weaknesses with the impacts of node capture attack.

In Ref. [7], the authors propose a practical model of deploying the sensors in groups. Here, the authors consider deployment of sensor groups in such a way that the same group members stay close to each other after the deployment in the network. Based on the deployment model, the authors develop a novel group-based key predistribution framework, which can be combined with any of the existing key predistribution techniques.

In Ref. [8], a group rekeying scheme is presented for broadcast security of a location aware WSN exploiting the relative location information of nodes to construct secure channels among the sensors and using those channels, the base station delivers the group key to each node in the network through a selected gateway node.

In Ref. [9], three types of nodes are defined: sensor nodes, base station nodes, and process center. Based on self-organizing clustering techniques, nodes are organized into different groups. Every group has one base station node and several sensor nodes. The base station node uses predeployed hash function to compute communication keys between different kinds of nodes in the network. Therefore, it can decrease the communication cost of dispatching keys. Meanwhile, the key management scheme is based on the theory of combinatorial optimization and provides an approach to maintain security while members have changed in groups.

In Ref. [10], a secure, efficient, and authenticated group key agreement protocol for WSNs is proposed by using node-ID and bilinear pairings. In comparison with the previous group key management schemes for WSNs, this scheme can defend against passive and active attacks, against node compromised attack, ensure the backward and forward security, and improve network computing complexity.

Zhou et al. [11] propose a group-based key predistribution scheme, GKE, which ensures secure node-to-node communication between any pair of sensors. According to Ref. [8], GKE provides a number of advantages like; accommodating different deployment models, establishing unique

pairwise key regardless of sensor density or distribution, nearly resilient feature against node capture attacks and low communications overhead.

Considering in-network process such as data aggregation, we need to explore the way to build a secure group communication for WSN. Prigent et al. [12] present a user-friendly distributed approach to set up and maintain a secure long term community over a home *ad hoc* network. In their scheme, there is no central point to the community because: each device of the community considers itself as the central point that is, any device can introduce any other in its community provided that they can communicate, even over insecure links.

Singh [13] does a study on the membership management protocols for groups in WSNs. The author investigates various sorts of applications, different geographic distributions, and membership models relevant to sensor networks. In Ref. [14], a secure multicast group is established and a group key is distributed by mutually authenticating a group of devices over an open insecure wireless channel.

In Ref. [15], two centralized group rekeying (CGK) schemes are proposed for secure group communication in sensor networks. In the first scheme, the group controller first obtains a group identifier from the base station and then generates a random key as the group key. All the receivers observing the same event would send a join request to the group controller. The group controller authenticates the request and unicasts the group key encrypted by the pairwise key to the sensor nodes. But it requires n unicasts of messages to update the group key which may cause heavy traffic in the area when the group size is large. So the group key could be distributed through broadcasting.

Benaloh and Mare [16] propose a one-way hash function which satisfies a quasi-commutative property that allows it to be an accumulator. This property could be used for time stamping, building trust relationship between entities in many systems, and for solving variety of problems. We use the quasi-commutative property of one-way accumulator (OWA) to serve our purpose of secure group association management in HDSN.

In Ref. [17], the authors study a healthcare monitoring architecture structured by three network tiers that provide pervasive, secure access to wearable sensor systems and wireless sensor motes. Their group-based approach to collect and transmit data is shown suitable and efficient to deal with medical services needed in a hospital. To secure the data dissemination processes, the authors also propose some security mechanism to encrypt the confidential data within the network and thus protect the privacy of patient information with a group security approach.

In Ref. [18], the authors develop a scheme for key management and rekeying in WSN based on the self-organized structure, grid-loop. Based on the proposed grid-loop topology, they propose new algorithms for key management, i.e., forming grid-loops *via* Minimum Spanning Tree and forming group key, to provide an original scheme to the WSN for creating loop keys and their maintenance and renewing. The idea of grid-loop is shown to be efficient compared to other existing cluster-based network formation schemes to deal with security issues.

Kifayat et al. [19] address group-based key management for both static WSN and mobile sensor networks. In their work, they provide mathematical details of dealing with key management by using groupings within sensor networks. Though this work is not directly related with our work, we mention this as some of the concepts regarding formation of groups and security mechanisms are useful.

Juwei and Liwen [20] present heterogeneous WSN structure where two types of sensor nodes are strategically deployed over the target area. Based on the strategic deployment model, the authors propose a security scheme based on Shamir secret sharing scheme. The work shows that the group-oriented cryptography could work well in practical cases for heterogeneous WSNs.

Motivated by these works, in this paper, we first propose our network model where sensors of different groups participate together and then we present our approach of secure group association management in the network. We adopt OWA for testing the legitimacy of the group members (sensors) in a particular group in the network. The subsequent sections will present the details of various aspects of our approach.

3. BACKGROUND AND MOTIVATION

3.1. What is HDSN?

Typical "*heterogeneity*" in WSN is considered based on the capabilities of the sensors in the network or more specifically based on the memory, processing capability, energy level, sensing range, and transmission range of the radio [21,22]. These aspects are often related with each other. For example, transmission range of a sensor depends on the available level of energy. Larger transmission range requires more energy or *vice versa*. So, in most of the cases, the *heterogeneity* is defined considering the dissimilarities in the energy level, processing power, and transmission range. However, in our case this term means that the sensors are associated with different groups, based on their functional types and after deployment, they collaborate with one another in the same group for doing any assigned task for that particular group. This group concept is different than the traditional relatively smaller groups (i.e., portions of the network). Figure 1 shows an example scenario. In a particular HDSN, let us suppose that, the temperature sensing sensors (type 1 sensors) form a portion of the total network whose task is to sense and report the average temperature. Side-by-side, there are other types of sensor groups (or deployed groups, DGs) in the same area for other tasks like monitoring the seismic signals in that area (consisting of all the type 2 sensors in the figure), and a DG (composed of all the type 3 sensors) for sensing the acoustic signals, etc.

We term our model network as HDSN, where sensors of various functional capabilities form different network-wide

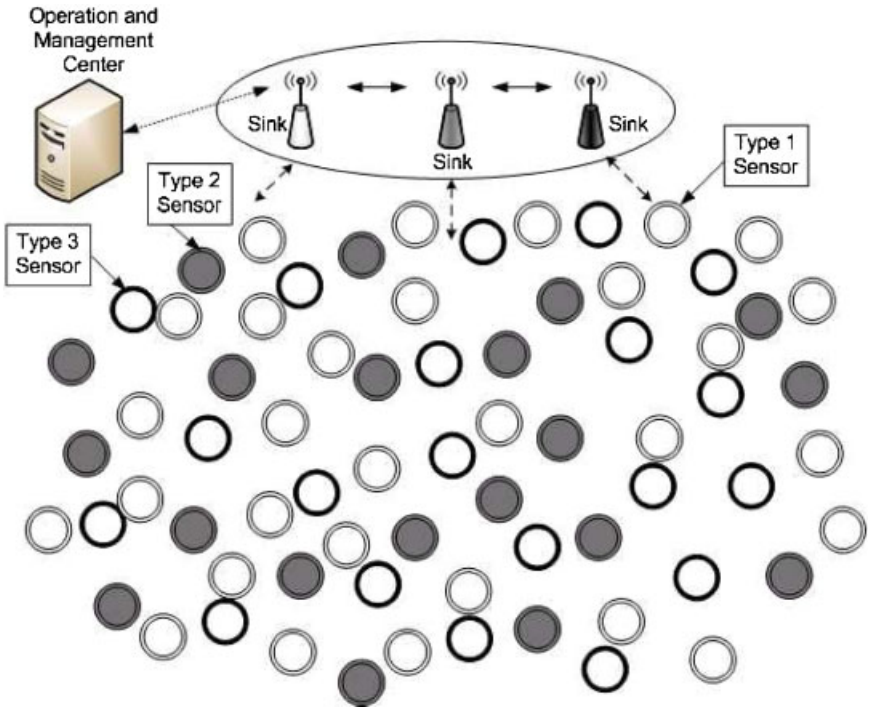

Figure 1. An example of HDSN deployment. Here, $T = 3$. Three types of sensors are dispersed over the same target area. The sensors with the same functional type (e.g., all the same colored sensors) are associated with the same network-wide DG.

DGs [23]. There could be T (where, $1 < T \leq 6$) functional types of sensors in the deployed network. We primarily consider at most six types of nodes in the same HDSN as this could be enough for supporting any of today's applications and many innovative multipurpose applications of sensor networks in the coming future. The value of T could also be set based on the application at hand. The network is called *homogeneous* only when $T = 1$, that means there is only one type of sensor in the whole network. For deploying the entire network, first a number of different types of sensors are taken and they are assigned different ids based on their functional types.

Figure 1 shows a graphical model of a HDSN. In this figure, $T = 3$. Like any other distributed sensor network (DSN) [24], it has a large number of sensors covering a large area. The deployment is dense so that a particular network region is covered perfectly. Also we assume that the sensors could frequently be added or deleted from the network. In the figure, we show three different sinks collecting data from three different DGs. A DG is composed of only one type of sensor and it spreads over the entire deployment area of the network. The sinks shown in the model are interconnected securely with each other. There could also be only one sink gathering all forms of data from different DGs and sending the readings to a operation and management center. In such a case, the processing burden of the sink increases as it has to collect, classify, and process different types of incoming data simultaneously.

Each of the DGs covers the whole area of interest (AOI) and works independently. However, the data packets from a sensor in one DG could be relayed by the sensors under another DG. So, practically in this sample HDSN, there are three DSNs of different functionalities that are working individually but side-by-side cooperating with each other for data transmissions and network operations. However, for collaborating for a particular task, the neighboring sensors must be the members of the same deployment group. That means even if two dissimilar sensors are neighbors to each other, they can only help in forwarding each other's packets but cannot take part in the same collaborative task. Figure 2 illustrates this. In the figure, three nodes 1, 2, and 3 are neighbors of each other but each of them is from a different network-wide DG. They can just relay each other's packets if needed (details are presented in Section 6.2.4), but cannot participate in the same task.

3.2. Why HDSN?

An important point of argument can be that; instead of using multiple nodes over the same area, similar data could be achieved by using multipurpose nodes (like ExScal motes) in the network. However, with the higher requirements of the type and accuracy of data, we have found that our approach is more efficient than the use of large number of homogeneous multipurpose nodes. For example, in a

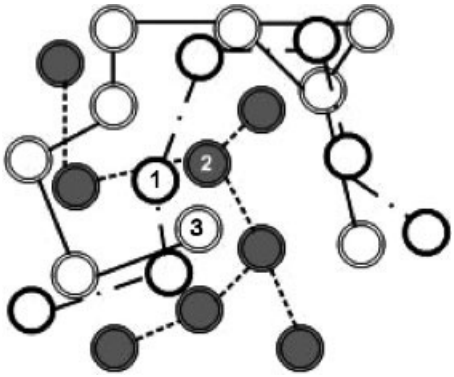

Figure 2. A portion of a HDSN. Nodes in different DGs are neighbors of each other but are not able to take part in the same collaborative task. For a given collaborative task, two neighbors must be of the same functional type. The distinguishable connecting lines show the achievable associations between any pair of nodes of same type.

volcano monitoring or wind tunnel monitoring application, utilization of our approach could obtain more accurate, and comprehensive information. Some sample cases where HDSN could be a preferable choice over other options are:

(i) Using our approach could be more efficient for saving energy. Firstly, in HDSN, a particular DG could be kept in sleep mode whenever necessary while other DGs can keep functioning. Furthermore, at least one DG could be preserved for a long time even if the energies of other types of nodes are exhausted. These two features can help for maximizing the lifetime of the overall network without having an influence on the connectivity of the whole network because of the dense deployment of sensors. On the other hand, ExScal type nodes need continuous wake mode when any of the sensing units on the same radio board is functioning. This eventually causes continuous consumption of energy resource (i.e., battery). Putting this type of multipurpose node in sleep mode means all of the sensing units would become idle at the same time.

(ii) Using our approach could be more efficient for improving security of the overall network. HDSN offers a great advantage over multipurpose node-based network in case of physical capture attack. If a physical intruder intends to hamper the sensing of different parameters in a particular location, it needs to destroy all types of sensors in that location. For example, in our shown model, the attacker needs to destroy at least three different type sensors from a particular section of the network. On the other hand, for a network consisted of multipurpose nodes, destroying one node is enough to destroy three sensing units assigned for a particular spot. The latter is a relatively easier task for a physical intruder. In addition, different security levels could be set for different DGs based on the requirements and/or functions of the sensors.

(iii) In HDSN, the burden of cluster heads could be reduced as different sinks associated with different DGs can gather and analyze specific type of data separately. They can even collaborate with each other after extracting the gist from the collected data. This facility is not available if ExScal type motes are used with a single sink. In many cases, multiple types of readings from multipurpose nodes can somewhat increase the complexity of tasks of the sink as it has to classify and reorder incoming data prior to manipulating them. Use of multiple sinks in such case cannot even help as various types of data packets are amalgamated in the incoming traffic for each sink.

As a whole, HDSN can provide more benefits that traditional sensor networks cannot. Taking such type of network setting into consideration, in this paper, we propose an approach of secure group association management within the network. As our focus is on group association security management, dealing with other aspects of HDSN is beyond the scope of this paper. After all these introductory texts, in the next two sections, we present the assumptions and preliminaries for our approach.

4. NETWORK ASSUMPTIONS

4.1. Network structure

We assume that in each DG, for each participating node, there is an end-to-end path from the corresponding sink. That means in a particular DG G_i, $i = 1, 2, 3 \ldots .T$ (where T is the maximum number of DGs in the HDSN), for each node n_{G_i}, there is a path from the corresponding sink, $S_{G_i} \rightarrow n_{m_i} \rightarrow (n_{m_i} - 1) \rightarrow \cdots n_{G_i}$. We adopt a scheme like that is presented in Ref. [25] for energy-efficient logical structuring of the network to get a sink rooted tree (SRT) for each DG. In this case, n_{G_i} is the leaf node and there could be zero or more intermediate nodes along the end-to-end path.

We assume that each sink associated with each DG has enough processing power to do the initial calculations to initiate the network-wide groups in the network. The sensors deployed in the network have the computational, memory, communication, and power resources like that of the current generation of sensor nodes (e.g., MICA2 motes [3] or MICA2DOT motes). Once the sensors are deployed over the target area, they remain relatively static in their respective positions; that means the neighboring nodes move together (if mobility is allowed) or do not move at all. The transmissions of each node are isotropic (i.e., in all directions) so that each message sent is a local broadcast within the transmission range of the node.

Based on these network assumptions, our goal here is to propose a mechanism by which the nodes in a particular DG

might securely recognize one another so that any adverse entity cannot in any way be included in the network within its operation time. This kind of group association verification could especially be required for performing some collaborative tasks in the network. For example, when the sink wants to know the average temperature of a certain region, the temperature sensors in that region might have to work together to measure the average of the temperature readings over that particular area. There could also be some sort of data aggregator or preprocessor which would be responsible for doing the primary calculations. In fact, our secure group association verification mechanism could also be used with other clustering mechanisms where there are some cluster heads present to collect these types of readings from a certain set of sensors.

4.2. Security assumptions and threat model

Assumption 1. There is an IDS that reports about any suspicious behavior of nodes in the network.

Assumption 2. The base station is physically secure and cannot be compromised in any way.

Due to the use of wireless communications, the nodes in the network are vulnerable to various kinds of attacks. We assume that an adversary could try to eavesdrop on all traffic, inject false packets, and replay older packets. The nodes are not tamper-proof. If in any case, a node is compromised, it could be a full compromise where all the information stored in that particular sensor is exposed to the adversary or could be a partial compromise, that is; partial information is exposed. Full compromise means that the adversary could use the secret information, cryptographic keys, sensor readings, etc. for facilitating its own purpose. Finally, we assume that it is possible that an attacker places malicious nodes in the network to get into the network for participating in the collaborative tasks or a set of such nodes can work together for colluding against a legitimate group of nodes (DG or subset of a DG) in the network.

5. BUILDING BLOCKS OF OUR SECURITY MANAGEMENT SCHEME

In this section, we introduce OWA and pseudoinverse matrix which are used as the building blocks of our group association security management approach.

5.1. One way accumulator

From the definition of a one-way function we know that it is a function F with the property that; for a given x it is easy to compute $y = F(x)$. However, given F, y, it is computationally infeasible to determine x such as $x = F^{-1}(y)$. Generally, one-way functions take a single argument. However, Benaloh and Mare [16] considered hash functions which take two arguments from comparably sized domains and produce a result of similar size. In other words, according to Ref. [16], a hash function is a function F with the property that, F: $A \times B \rightarrow C$ where, $|A| \approx |B| \approx |C|$. This view introduces the one-way hash function with a special *quasi-commutative* property which is termed as OWA. According to the definition, OWA is a one-way function, f: $X \times Y \rightarrow X$ with the *quasi-commutative* property such that, for all, $x \in X$ and for all, $y_1, y_2 \in Y$,

$$f(f(x, y_1), y_2) = f(f(x, y_2, y_1))$$

A family of OWAs is a family of one-way hash functions, each of which is *quasi-commutative*.

This property is not unusual. In fact, addition and multiplication modulo n both have this property as does exponentiation modulo n when written as, $e_n(x,y) = x^y \bmod n$. Modular exponentiation also satisfies the *quasi-commutative* property of OWA:

$$f(f(x, y_1), y_2) = f(f(x, y_2), y_1) = x^{y_1 y_2} \bmod n$$

This could be extended for a long sequence of y_j values (where, $j = 1, \ldots, m$).

The *quasi-commutative* property of OWAs f ensures that if one starts with an initial value, $x \in X$, and a set of values $y_1, y_2, \ldots, y_m \in Y$, then the accumulated hash,

$$z = f(f(f(\cdots f(f(f(x, y_1), y_2), y_3), \cdots, y_{m-2}), y_{m-1}), y_m)$$

would be unchanged if the order of the y_js were permuted. This feature could be used for membership verification in a large set of entities. We adopt this feature of OWA for secure group association authentication in HSDN.

5.2. Pseudoinverse matrix

The pseudoinverse matrix or generalised inverse matrix [26–28] has a very nice property that could be used for cryptographic operations. It is well known that a nonsingular matrix over any field has a unique inverse. For a general matrix of dimension $k \times w$, there might exist more than one generalized inverse. This is denoted by, $M(k, w) = \{A{:}A$ is a $k \times w$ matrix$\}$. Let $A \in M(k, w)$. If there exists a matrix $B \in M(w, k)$ such that; $ABA = A$ and $BAB = B$ then each of A and B is called a generalized inverse matrix (or pseudoinverse matrix) of the other. In this paper, we use the notation A_g to denote the generalized inverse matrix of A. We use pseudoinverse matrix for the pairwise key derivation process presented later in the paper.

It should be noted that $(A_g)_g = A$ is not always true. The set of all possible pseudoinverse matrices of A is denoted by $\{A_g\}$ and $|\{A_g\}|$ is the cardinality of $\{A_g\}$. Then, we have:

Lemma 1 Let A_g be a pseudoinverse matrix of A. Then, $\text{rank}(A_g) = \text{rank}(A)$.

Lemma 2 Let $A \in M(k, w)$with $rank(A) = k$. If A can be written as $A = [A_1, 0]$, where A_1 is a k × k nonsingular matrix then,

$$\{A_g\} = \left\{ \begin{bmatrix} A_1^{-1} \\ Z \end{bmatrix} : Z \in M(w-k, k) \text{ is an arbitrary matrix} \right\}.$$

Proof Let $B = \begin{bmatrix} X \\ Z \end{bmatrix} \in M(w, k)$. It is then easy to verify that both $ABA = A$ and $BAB = B$ hold if and only if $X = A_1^{-1}$.

6. SECURE GROUP ASSOCIATION MANAGEMENT IN HDSN

6.1. Naive approach

The naive approach to maintain the group association information of a particular group of sensors could be storing the member ids in each sensor node's memory. However, the storage requirement for such member id list linearly increases with the increase of the number of sensors in that particular group. For the sensors with limited storage capabilities, this is not a good solution. Hence, we employ an efficient OWA-based scheme for managing the membership information of a group in such a way that it could well be supported by the storage and computation power of the modern-era sensors. Also based on this limited information, the sensors in a particular network-wide DG can securely verify and recognize each other.

6.2. Our approach: based on one-way accumulator and pseudoinverse matrix

6.2.1. Calculating partial accumulated hash value (PHV).

In case of OWA, if the values $y_1, y_2, \cdots, y_m$ are associated with the users of any cryptosystem, the accumulated hash z of all of the y_js can be computed. A user holding a particular y_j can compute a partial accumulated hash z_j of all y_j with $i \neq j$. The holder y_j can then demonstrate that y_j was a part of the original hash by presenting z_j and y_j such that; $z = f(z_j, y_j)$. We use this partial accumulated hash values (PHV) in the group association verification process. The following subsections present our scheme in details.

6.2.2. Preprocessing and prestoring of PHVs.

Before deployment of a group of sensors (i.e., a network-wide DG), the following steps are performed:

(i) A unique id; y_j, $j = 1, \cdots, m$ is assigned for each sensor participating in a particular deployment group.
(ii) Two safe relatively prime numbers, p and $q = 2p + 1$ are generated.
(iii) n and $\phi(n)$ are computed as; $n = pq$ and Euler's totient function, $\phi(n) = (p-1)(q-1)$.
(iv) A random number x (as a seed) is generated which is same for every node in the group.
(v) *PHV* for each node y_j is computed using the formula,
(vi) $z_j = x^{\prod_{i=1, i\neq j}^{m} y_i} \bmod n$
(vii) Now the values of z_j, n, $\phi(n)$, and corresponding y_j are stored in each sensor in that particular deployment group.

6.2.3. Post-deployment secure group association verification.

After deployment of the sensors in the AOI, if a node needs to verify the association of another node (whether they are in the same DG or not), the PHVs and the identities of the nodes are used. For example, let us suppose that two nodes n_p and n_q want to verify whether they are in the same group or not. For this membership verification, these two nodes exchange their prestored PHVs z_p, z_q and their identities, y_p and y_q. Node n_q calculates $z = f(z_p, y_p) = {z_p}^{y_p} \bmod n$, while the other node calculates, $z = f(z_q, y_q) = {z_q}^{y_q} \bmod n$ locally. If both of the locally computed OWA values match with each other, the nodes could be sure that they are participating as the siblings in the same DG in the HDSN. Once the accumulator value is calculated and matched, it could be preserved in the node for successive node membership verification for a given collaborative task.

6.2.4. Secure pairwise key derivation between two sensors of two different DGs.

Since for each DG, there is a SRT (as mentioned in the network assumption in Section 4), once a node in a particular DG finds and verifies its siblings, it can use them for forwarding even its own readings (alongside the results of collaborative tasks) towards the sink. The problem arises when a node is disconnected or misplaced from all other nodes of its own DG. It can happen due to the failure of an intermediate node of the same DG, or because of random (or, poor) deployment of the sensors of the same types. In such a case, though the stranded node cannot participate in the collaborative tasks within its area, it might need to send its own readings to the sink/base station. In fact, it can even happen for a subset of nodes associated with a particular DG. To handle this issue, our mechanism uses a simple method of deriving pairwise keys [28] between two neighboring sensors even if they are associated with two different DGs. Here we describe the secure pairwise key derivation method.

Let us consider that n_{G_X} and n_{G_Y} are two nodes in the deployment groups G_X and G_Y. For some reason, n_{G_X} has been disconnected from its siblings but it has got n_{G_Y} as its neighbor, which is connected with the SRT of its own DG (i.e., G_Y). To derive a shared secret key between these two nodes, following operations are performed:

(i) Node n_{G_X} randomly generates a matrix X with dimension $m \times w$ and its psedoinverse matrix, X_g. These matrices are kept secret in the node.
(ii) n_{G_X} Calculates $X_g X$ and sends it to n_{G_Y}.
(iii) In turn, N_{G_Y} randomly generates another matrix Y with dimension $w \times k$, and finds out its pseudoinverse matrix Y_g. These matrices are also kept secret in node N_{G_Y}.
(iv) N_{G_Y} Calculates $X_g XY$ and $X_g XYY_g$. Then it sends the resultant matrices to n_{G_X}.
(v) Upon receiving the products of matrices from n_{G_Y}, n_{G_X} computes, $XX_g XYY_g = XYY_g$ and sends it back to n_{G_Y}.

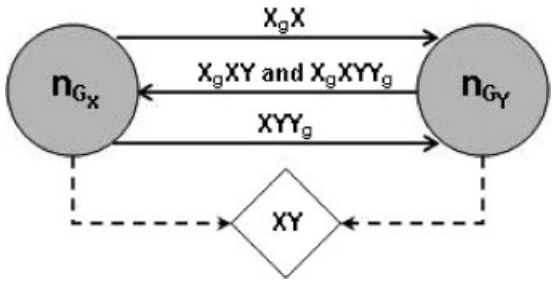

Figure 3. Pairwise key derivation process between two nodes.

(vi) Now, both the nodes $n_{G_X}$and n_{G_Y}can compute the common secret key. n_{G_X} gets it by calculating $X(X_gXY) = XY$ and n_{G_Y} gets it by calculating$(XYY_g)Y = XY$. Both of these outcomes (XY) are the same matrix with dimension $m \times k$.

Figure 3 shows the communications between the two neighboring sensors in the pairwise key derivation process. Basically, the key XY is locally computed by each node and the entire method is executed without the intervention of any third party. The derived pairwise key now could be used for secure communications between two nodes. In our case, the node n_{G_X} encrypts all its readings using the shared key (XY) and sends them to n_{G_Y} which in turn uses its own DG's (i.e., G_Y) SRT to forward those readings to the sink. Such readings are sent with special marks in the packets so that the corresponding sink can recognize the irregular packets and hand those over to the appropriate sink (see Figure 1). In this way, the readings of a stranded node could be utilized and two neighboring nodes associated with two separate DGs can cooperate with each other for secure data handover within the network.

Figure 4 shows an example scenario. In this figure, node n_s of type 3 is stranded from all of its other legitimate siblings. It has got only type 1 and type 2 sensors as its neighbors. In this case, node n_s derives a pairwise key with n_f and forwards its readings to the sink/base station *via* node n_f (of type 1). If any collaborative task is assigned for the type 1 sensors in that particular deployment region, n_f would work with other type 1 sensors avoiding n_s as it is not a legitimate member of its own DG (i.e., DG of n_f). However, because of the request of n_s (special case here), it forwards the readings using its own DG. After receiving the specially marked packets, the sink associated with n_f hands them over to the appropriate sink originally associated with n_s. We term such a stranded node (n_s), an *orphan*. The immediate forwarder node (i.e., n_f) of different DG is termed as the *step-brother* of the *orphan*.

6.2.5. Addition of new member in a deployment group.

Addition of new sensors in a particular DG could be handled in two ways. At the time of deploying the sensors, all the sensors assigned for a group might not be used, rather

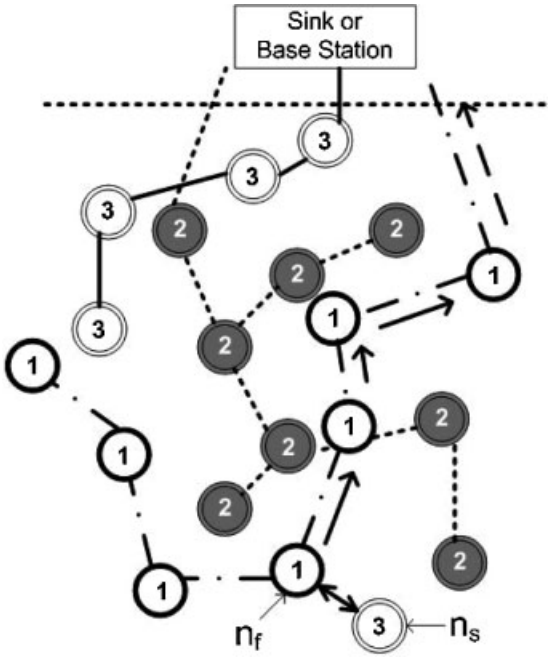

Figure 4. Delivery of data using an end-to-end path of a different DG. Node n_s and n_f derived a pairwise key for their communications even though they are from different DGs. However, they cannot partake in any collaborative task assigned for the particular deployment region. Here, n_s is an orphan and n_f is the step-brother of n_s.

some of them could be kept for later use (depends on the nature of application at hand). Say for example, total number of sensors in a group before deployment is λ. So, a certain portion, say η of these sensors could be deployed first for that particular group and the remaining, $(\lambda - \eta)$ sensors could be added later in the network. In such a case, all the newly added sensors could still be able to prove their legitimacy of group association to other already deployed sensors in that group using our OWA-based verification scheme. This way of addition of new nodes is basically a policy-based management approach, where we handle the addition of new sensors by employing a good deployment strategy.

However, OWAs allow addition of completely new sensors for a certain DG in the HDSN. For example, let us consider that a new sensor has the id y_{new}, which is assigned from the base station. Mathematically, the new OWA is, $z_{\text{new}} = f(z, y_{\text{new}})$. To inform all the sensors in that particular DG about the newly added sensor, the base station uses the dedicated end-to-end path (according to our assumption) of each sensor. In turn, each sensor updates its PHV using the formula,

$$z_{j_{\text{new}}} = f\left(z_j, y_{\text{new}}\right) = z_j{}^{y_{\text{new}}} \bmod n.$$

Before deployment of the new node, the base station calculates its PHV and stores it in its memory. To add a new set of sensors in a DG, the base station securely sends all the ids of the newly included sensors to the deployed sensors of

that deployment group. One special case exists when there are one or more *orphan*s in the network. In such case, as the *orphan*s are already cut off from their respective DGs, they cannot receive the update messages about sensor addition. Once a sensor gets an *orphan* status, it is not allowed to rejoin its own DG even if it gets newly deployed sensors of its own type as its neighbors (that could repair the broken connection with other sensors of the same type). This policy is used to avoid complexity of recalculating the actual PHV for the *orphan* as it might have missed several update messages by the time it again gets one of its siblings as its neighbor. So, for this case, the *orphan* just provides service as a self-sufficient node to supply its own readings. As it has already established a relation with other *step-brother*, it can keep that relation until it is out of battery. Note that this policy does not prevent other legitimate nodes in the SRT of the original DG from proper functioning. They could simply follow the calculations and remain in the group as legitimate entities.

6.2.6. Deletion of member from a deployment group.

Any suspicious behavior detected by the IDS could convince the base station (or, sink) to purge any sensor from a particular deployment group in the HDSN. To purge an adverse node y_{adv}, the sink uses the secure end-to-end paths for the sensors in the DG to send the id of the deleted node. Getting the delete command, each of the remaining nodes calculates the stored PHV using the equation,

$$z_{u\_j} = z_j^{y_{\text{adv}}^{-1} \bmod \phi(n)} \bmod n$$

Euler's totient function $\phi(n)$is used here for the modular operation to ensure that underflow does not occur and the purged id could not be reused by any adversary. In case of a node failure due to any unwanted incident like power outage (or other), it should be made sure that the node's id could not be used by any other entity or any attacker. So, to handle this, the same procedure for node purging is employed. In all of these cases, the sink is responsible for taking the decision of purging. In some applications, where the clustering techniques are employed, it is possible to assign the charge of taking group-related decisions to the cluster head of the particular cluster (or subgroup). In this case, our scheme offers a decentralized node membership verification mechanism and reduces the burden of tasks of the corresponding sink. For handling the presence of *orphan*s in the network, same policy as stated in previous section (Section 6.2.5) is employed.

7. PERFORMANCE ANALYSIS AND DISCUSSION

We analyzed our scheme in terms of storage requirements, computation costs, communication costs, security, and scalability. To understand the performance of our approach, in our system setting, we varied the number of nodes in the deployment groups from 100 to 1000. In this section, we present the analysis and discussion about the efficiency and applicability of our approach with current generation sensor nodes.

7.1. Storage requirements

As we use similar type of scheme like RSA [29], like RSA cryptosystem, our scheme requires that the size of n should be sufficiently large, typically 1024 bits or more. We considered different lengths of z_j, n, $\phi(n)$ with three different lengths of the ids of the nodes; 128 , 512 , and 1024 bits. Each sensor node in a particular deployment group has to store little information; only four values for the secure group association verification mechanism. Figure 5a,b shows the storage requirements for a single sensor when our scheme is employed considering different lengths for the stored parameters. In Figure 5a, lengths of z_j, n, $\phi(n)$ are considered as 1024 bits and in Figure 5b, they are 1280 bits. We took MICA2 mote as a standard specification. Crossbow MICA2 mote [3] is a well-known sensor node with an ATmega128L 8-bit processor at 8 MHz, 128 KB program memory (flash), and 512 KB additional data flash memory. Usually it is powered by 2 AA sized batteries. Consider-

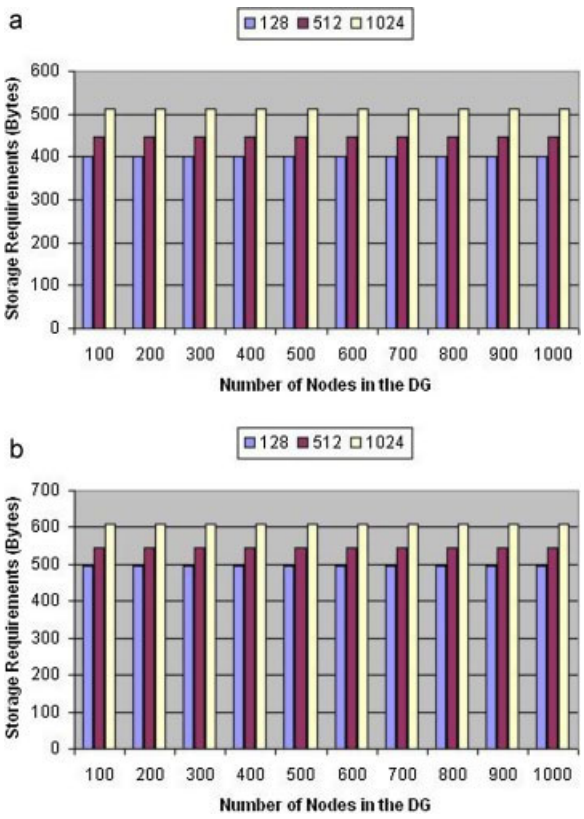

Figure 5. Storage requirements for different length node ids keeping the lengths of other parameters (a) 1024 bits (b) 1280 bits.

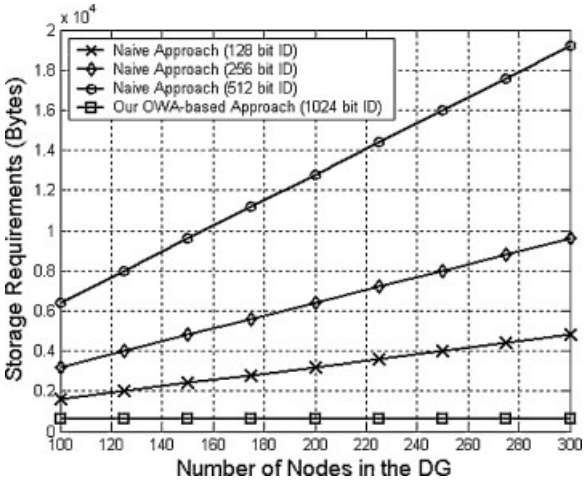

Figure 6. Storage requirement for a single node in naive approach (for different lengths of IDs) and our OWA-based scheme (considering ids 1024 bits and each of other values 1280 bits in length).

ing this configuration, our scheme requires only a small amount of memory while a large portion of memory remains available for other associated mechanisms to run smoothly.

An alternative method of group association management is to store all the ids of the sensors (naive approach) participating in the same group. But, compared to our scheme, this approach is very inefficient. In Figure 6, we show that if simple membership list is stored for the same purpose, it requires much more storage than our OWA-based approach. In fact, for a small number of nodes (100–300), the storage requirements are very high which could in fact hamper proper functioning of other schemes running alongside group association management scheme. While for different length ids, naive approach (storing membership list) requires huge amount of memory; in OWA-based approach, considering the lengths of other parameters even 1280 bits (which is good enough for the security) and id length 1024 bits, the memory requirement is fairly less. A great advantage of our approach is that the increase of the number of nodes in the DG does not affect the storage requirements for a sensor. Moreover, the pairwise key derivation method does not need any prior storing of information (predistribution) in the sensors' memories rather the scheme could be used whenever needed for handling *orphan*s. Hence, with our mechanisms, a large number of nodes can be supported for a HSDN. For larger lengths of n and other parameters, the storage requirement increases. However, still it is fairly affordable by today's sensors and comparatively much less than that of storing the whole membership list.

7.2. Computation costs

A great advantage of our approach is that; as the entire preprocessing phase (presented in Section 6.2.2) is done by the base station (or, management center), the sensors do not need to bother about the calculations and no extra sensor resource is used for initializing the deployable groups (DGs). The sensors need to use the processing resources only for verification, sensor addition, or sensor deletion process. We found that; in the verification step for calculating the accumulator value, a node takes only about 3 ms when 1024 bit id is used. This processing time is fair enough for such a scheme. Depending upon the size of the id, the processing time varies a little. In case of handling an *orphan*, to derive pairwise key, we have used linear matrix operations, more specifically matrix multiplication. The complexity of matrix multiplication is very low; hence it could be performed very quickly. As all the computations in this case are linear, they can be performed very easily. The point that should be mentioned here is that; having orphans in the network is the worst case. So the key derivation method might not even be needed if an efficient deployment strategy is used for deploying the sensors uniformly in the network. As we use the protocol presented in Ref. [25] for SRT maintenance, each DG could lose energy in an efficient way so that a DG remains connected before its full exhaustion and no orphan is created due to the power outage of the intermediate nodes.

7.3. Communications costs

In the post-deployment secure group association verification phase, only two message transmissions are needed between two neighboring nodes. This process is iterated among the neighbors to ensure the legitimacy of each other and the pairwise authentications could be sufficient to legitimize a good number of nodes of same type (or, a subset of nodes) in a particular deployment area.

Say for example, nodes A and B, two neighbors verified each other as siblings. C is a neighbor of B, but not a neighbor of A. B and C can verify each other's legitimacy. If all of them are of same type and any of them has not already got the orphan status, they (A, B, and C) can verify each other and then can work together for any assigned collaborative task. In this example case, if C is also a neighbor of A, they again can authenticate each other. Then, the whole subgroup becomes pairwise authenticated (shown in Figure 7). As the deployment is dense, there could be many such subgroups within the network. One of the nodes can take the role of the leader to accumulate the readings of the verified nodes for that collaborative task. The details of data aggregation and distributed manipulation of sensor readings are beyond the scope of this paper as here we focus on the group security management issues only. As the transmissions of the nodes are isotropic, the neighbors can communicate within their transmission circles and these could be done with usual neighbor communications methods. When sensors are added in the network or deleted from the network, the update messages need to be sent throughout the SRT of a particular DG. These communications are done using the scheme we adopted from Ref. [25]. So, as the information about addition/deletion is transmitted as a part of that scheme, there is no extra communication

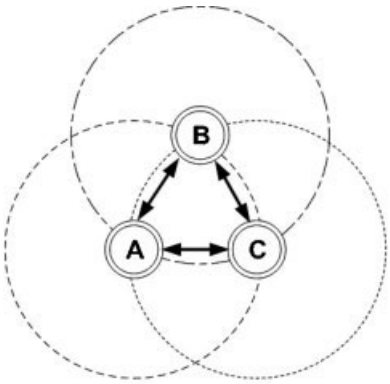

Figure 7. Example scenario of a pairwise authentication case. Nodes A, B, and C authenticated each other using the OWA-based verification mechanism and all of them are of same type. Simple neighbor messages could be used to exchange PHVs and ids of the nodes. All these nodes could partake in a collaborative task. Dotted circles show the transmission ranges of the nodes.

needed to support our secure group association management scheme.

If there are orphans in the network, pairwise secret key derivation is needed. In this scheme, total number of transmissions needed is three. The sending entity sends an $w \times w$ matrix which is of w^2 bits. In turn, the receiving entity sends a $w \times k$ matrix and an $w \times w$ matrix. For this the total number of bits passed for the matrices is, $w^2 + wk = w(w + k)$ bits. Again, the sending entity sends the receiving entity $m \times w$ bits. So, total number of bits (for the matrices) needs to be exchanged in this method is,

$$w^2 + w(w + k) + mw$$

$$= w(w + w + k + m)$$

$$= w(2w + k + m) \text{ bits.}$$

7.4. Security analysis

7.4.1. Analysis on one way accumulator.

Now, let us analyze the security level of our schemes. One-way accumulator uses one-way hash function which means that; given $x \in X$and $y \in Y$, for a given $y' \in Y$, it is difficult to find some $x' \in X$ such that; $f(x, y) = f(x', y')$. So, an adversary that wants to forge a particular y' would face the difficulty of constructing an x' with the property that; $z = f(x', y')$. Likewise, in our scheme, the use of arbitrary values for PHV and identity of node cannot pass the group association verification mechanism and the adversary cannot in any way be included in the deployment group even if it is of same functional type. A potential threat is that if a dishonest member in the group tries to construct a false pair (x', y') such that,$z = f(x', y')$ by combining various node identities (y_js) in one-way or another. However, as mentioned earlier, this is not practical as the adverse node faces the difficulty of finding such a pair. Other methods of generating the pair might be possible. However, this could be handled by restricting the choice of the identities (set of y_js) of nodes, which is dependent on the decision of the central entity and based on the application requirements and/or network settings.

In the preprocessing stage, we use a rigid value of n. According to Benaloh and Mare [16], the advantage of using a rigid integer, $n = pq$ is that the group of squares (quadratic residues) modulo n that are relatively prime to n has the property that it has size, $n' = ((p-1)/2)/((q-1)/2)$ and the function, $e_n(x, y) = x^y \text{mod} n$ is a permutation of this group whenever y and n' are relatively prime. Thus, if the factorization of n is hidden, "random" exponentiations of an element of this group are extremely unlikely to produce elements of any proper subgroup. This means that repeated applications of $e_n(x, y)$ are extremely unlikely to reduce the size of the domain or produce random collisions. Although constructing rigid integers is somewhat harder than constructing ordinary, "*difficult to factor*" integers, it is still quite feasible.

7.4.2. Analysis on pseudoinverse matrix.

In the pairwise secret key derivation method, we use the public channel for the message transmissions. However, capturing the messages like X_gX, X_gXY, X_gXYY_g, and XYY_g could not be helpful to construct the locally computed secret shared key XY. It might seem that a prospective attack would be by gaining some information of matrix Y from the knowledge of X_gX and X_gXY. Let us consider, $\text{rank}(X) = r$. Let us assume that,

$$X_gX = \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix}.$$

Here, I_r is an identity matrix of order $r \times r$. Then, only the first r rows of Y can be determined from X_gXY. As Y is chosen randomly, there are $2^{(w-r)k}$ ways to choose the last (w–r) rows of Y. The knowledge of X_gXYY_g might be helpful in determining Y but according to lemma 2 it does not help much to the adversary. Without the knowledge of last (w–r) rows of Y, even if X is completely known, the probability for determining the correct value of each element of XY would be 0.5, considering that any row vector of X has a nonzero element in any of the last (w–r) positions, which is likely when (w–r) is considerably large and X is chosen at random. However, the possibility of getting X from the knowledge of $X_gXYY_g$and XYY_g is even smaller as $\text{rank}(A_gABB_g) \leq \text{rank}(A_gA)$. So, it could be assumed that based on the above analysis of the pairwise key derivation phase, the probability of successful breaking of this scheme is, $2^{-(w-r)k}$. So, the security of the pairwise key derivation scheme is reasonably high for carefully chosen parameters. To ensure that $2^{(w-r)k}$ is a large number, the w for this scheme must be considerably larger than r. And this can be guaranteed by making sure that $m < w$.

A potential attack could arise in the pairwise key derivation process if there exists any sort of identification problem

of the participating entities during the communications. In our scheme, however, this threat is debilitated to an acceptable level because; firstly, the presence of *orphan*s and thus the utilization of pairwise key derivation method is not likely if a good deployment policy is used where the sensors of a particular type are deployed uniformly and densely. Secondly, even if an adversary pretends to be an *orphan* and establishes key with a *step-brother*, it can only do harm by injecting huge number of false packets. As it cannot participate in the collaborative tasks, it cannot extract information from the network and in fact, it has to work more than the benefit it can get from such activities. Thirdly, even if the adversary pretends to be an *orphan* and derives a pairwise key, this key does not have any other use except using it for delivering the generated false readings. Finally, if the first packet reaches the sink along the path of the *step-brother*'*s* SRT (or, for its DG), the sink can check the id of the source of the specially marked packet and as it has the complete list of authentic members of a group, it can easily detect the falsehood and block further reception of packets from such source. This operation is basically a part of the IDS that runs side-by-side our approach. The IDS also filters out the spoofed ids of sensors. Details of the IDS and its methods of operations are left as our future work and will be noted in our future publications.

In fact, such activity by an adversary requires it to be an insider in the network which would not be allowed by the efficient IDS present in the network. If the adversary generates packets as an outsider, the sink will do the rest for stopping/blocking its production of false readings as stated earlier. As a whole, the probability of such attack is very low and even if it occurs, the level of damage from such attack is fairly manageable.

7.4.3. Analysis on intrusion detection system (IDS).

A potential attack against HDSN could be network substitution attack [30]. In such kind of attack, the adversaries take control of the entire network or a portion of it using a set of colluding malicious nodes. Once the rogue nodes are somehow included in the SRT of a DG, they can launch collusion attack. If that particular portion of the network is chosen for a collaborative task, all these colluding nodes can generate false reports. Also, when the adversaries control a portion of the network, they can perform other attacks such as traffic analysis and selective or complete packet dropping. However, in our secure group association management scheme, such types of attacks could easily be foiled. The IDS present in the network would work for detecting the presence of rogue nodes. Even if a number of colluding nodes are included in the SRT of a particular DG, they must know the information prestored in the legitimate sensors' memories. If PHV value and legitimate id are not known, none of the rogue nodes can produce proof of its authenticity to a legitimate member and thus cannot participate in the collaborative task.

If a legitimate node of a DG is fully compromised, all the prestored values might be known to an attacker. In such case, the attacker could use the information for including itself in any collaborative task. However, the extent of damage from such type of insider attack could be less if a single node is compromised as it cannot alter the result of the collaborative task as the readings from several sensors are considered together. Only way the attacker can forge the result is by producing outlier (an extreme deviation from mean). It is expected that detecting such extreme deviation in sensor readings is one of the responsibilities of the distributed IDS employed in the network. If a good number of nodes are fully compromised and their data are used for network substitution attack, this is the worst-case scenario. The burden of detection and exclusion of the rogue nodes again lies on the IDS that we have left as our future work.

7.5. Energy analysis

As we have mainly focused on the novel deployment model of the network, only some portions need the energy analysis. For the pairwise scheme, we have analyzed the energy consumption and found that our scheme could be easily supported by the modern day sensor nodes. For energy analysis, we considered the specifications of Berkeley/Crossbow MICA2DOT motes (a version of MICA2) [31]. These motes are equipped with eight-bit ATmega128L microcontrollers with a 4 MHz clock speed, 128 kB program memory and Chipcon CC1000 low-power wireless transceiver with a 433–916 MHz frequency band. The major power consumers in this mote are the processor and the wireless transceiver. During the transmission and reception operations, the microcontroller is turned on along with the wireless transceiver. According to our findings, the cost of transmission of one byte is 59.2 μJ, while the reception operation has about half the transmission cost (28.6 μJ). The power to transmit one bit is equivalent to roughly 2090 clock cycles of execution of the microcontroller. In our calculation, we considered a packet size of 41 bytes (payload of 32 bytes, header 9 bytes). With an eight byte preamble (source and destination address, packet length, packet ID, CRC, and a control byte) for each packet; we found that to transmit one packet $49 \times 59.2 = 2.9008 \approx 2.9$ mJ energy is required. Accordingly, the energy cost for receiving the same packet is $49 \times 28.6 = 1.4014 \approx 1.4$ mJ. Overall, the scheme is within the energy resource budget of current-generation sensor nodes.

7.6. Scalability

Our approach is fairly scalable. As the number of nodes in a DG does not affect the amount of memory needed for a DG verification method, a large number of nodes could be initialized before deployment. Also, new sensors could be added, if necessary with the same low memory requirement.

This particular advantage of memory efficiency also helps our approach to be fairly scalable.

7.7. Further discussion

In this paper, we have focused on ensuring secure group association management for the DGs in HDSN. Other security mechanisms can run side-by-side our schemes. In our approach, several groups of sensors could operate in the same HDSN at the same time without hampering each other's operations. If required, there could be some other mechanisms for the communications between the sinks of different groups or the nodes of different groups. This actually depends on the type of service required from the HDSN or the application at hand. At the time of deployment, if policy-based deployment is used, a certain portion of the sensors could be kept for future deployment. Yet, it would not affect anything in the deployed sensor group and all the other sensors in the DG can still verify each other's legitimacy of membership.

8. CONCLUSIONS AND FUTURE SCOPES OF RESEARCH

The major contributions of this paper are as follows:

(i) We have proposed a new deployment model of DSN termed HDSN.
(ii) Based on the novel deployment model, we then proposed a secure group association management scheme. Our scheme could be employed alongside other supplementary security mechanisms for HDSN.
(iii) We also have presented a pairwise secret key derivation method between two different sensor nodes. Our analysis shows that this approach requires considerably very small storage and processing power, and is efficient enough to ensure secure membership of nodes in the deployment groups in HDSN.

Our work opens the door for research on other interesting issues in HDSN. For example, handling heterogeneous traffic, prioritized data, maintaining heterogeneous levels of security, quality of service of heterogeneous data, lifetime maximization of deployment groups, etc. could be some of the challenging research issues for HDSN. As our future work, we will develop an efficient distributed IDS that could run alongside our approach for ensuring robust security in this type of wireless network.

REFERENCES

1. Gu L, Jia D, Vicaire P, *et al*. Lightweight detection and classification for wireless sensor networks in realistic environments. In Proceedings of ACM SenSys 2005, 2–4 November, San Diego, California, USA, pp. 205–217. 2005.
2. Dutta P, Grimmer M, Arora A, Bibyk S, Culler D. Design of a Wireless Sensor Network Platform for Detecting Rare, Random, and Ephemeral Events. In Proceedings of the 3rd symposium on Information Processing in Sensor Networks (IPSN'05), LA, California, pp. 497–502. 2005.
3. https://www.eol.ucar.edu/rtf/facilities/isa/internal/CrossBow/DataSheets/mica2.pdf [05 October] 2010].
4. Chan H, Perrig A, Song D. Random key predistribution schemes for sensor networks. Proceedings of the 2003 IEEE Symposium on Security and Privacy, p. 197. 2003.
5. Liu D, Ning P. Location-based pairwise key establishments for static sensor networks. ACM Workshop on Security of Ad Hoc and Security of Ad Hoc and Sensor Networks, pp. 72–778. 2003.
6. Canh NT, Lee Y-K, Lee SY. Hgkm – a group-based key management scheme for sensor networks using deployment knowledge. In Proceedings of Sixth Annual Conference on Communication Networks and Services Research, May 2008, pp. 5–8.
7. Liu D, Ning P, Du W. Group-based key predistribution for wireless sensor networks. *ACM Transactions on Sensor Networks* 2008; **4**(2): Article 11.
8. Biswas S, Afzal SR, Lee G, Kim D-k. A Group Rekeying Scheme for Location-aware Sensor Networks. In Proceedings of International Conference on Information Security and Assurance, pp. 276–279. 2008.
9. RuiYing D, HuiJuan T, Song W. An Efficient Key Management Scheme for Secure Sensor Networks. In Proceedings of the Sixth International Conference on Parallel and Distributed Computing Applications and Technologies, pp. 279–283. 2005.
10. Zhang Z, Jiang C, Deng J. A Novel Group Key Agreement Protocol for Wireless Sensor Networks. Proceedings of 2010 International Conference on Measuring Technology and Mechatronics Automation, vol. 1, pp. 230–233. 2010.
11. Zhou L, Ni J, Ravishankar CV. Efficient Key Establishment for Group-Based Wireless Sensor Deployments. In Proceedings of the 4th ACM WiSE'05, Cologne, Germany, pp. 1–10. 2005.
12. Prigent N, Bidan C, Andreaux J-P, Heen O. Secure Long Term Communities in Ad Hoc Networks. Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'03), pp. 115–124. 2003.
13. Singh KH. "A Study of Membership Management Protocols for Groups in Wireless Sensor Networks," M.S. thesis, Dept. Computer Science, University of Illinois at Urbana-Champaign, USA, 2004.
14. Ghosh SK, Patro RK, Raina M, Thejaswi C, Ganapathy V. Secure group communication in wireless sensor

networks. First International Symposium on Wireless Pervasive Computing, Phuket, Thailand, 2006.
15. Wang Y, Ramamurthy B. Group rekeying schemes for secure group communication in wireless sensor networks. *IEEE International Conference on Communications'07* 2007; 3419–3424.
16. Benaloh J, Mare Md. One-way Accumulators: A Decentralized Alternative to Digital Signatures LNCS. vol. **765**. Springer-Verlag: 1994; 274–285.
17. Huang YM, Hsieh MY, Chao HC, Hung SH, Park JH. Pervasive, secure access to a hierarchical sensor-based healthcare monitoring architecture in wireless heterogeneous networks. *IEEE Journal on Selected Areas in Communications* 2009; **27**(4): 400–411.
18. Zeng Y, Xia Y, Su J. A new Group Key Management Scheme based on DMST for Wireless Sensor Networks, In Proceedings of the IEEE 6th International Conference on Mobile Adhoc and Sensor Systems 2009 (MASS'09), pp. 989–994. 2009.
19. Kifayat K, Merabti M, Shi Q, Llewellyn-Jones D. Group-based Key Management for Mobile Sensor Networks. In Proceedings of 2010 IEEE Sarnoff Symposium, Princeton, NJ, USA, 2010. pp. 1–5.
20. Juwei Z, Liwen Z. A Key Management Scheme for Heterogeneous Wireless Sensor Networks Based on Group-oriented Cryptography, 2010 International Conference on Internet Technology and Applications, Wuhan, China, pp. 1–5. 2010.
21. Ai C, Hou H, Li Y, Beyah R. Authentic delay bounded event detection in heterogeneous wireless sensor networks. *Ad Hoc Networks*, 2009; **7**, (3): 599–613.
22. Mhatre VP, Rosenberg C, Kofman D, Mazumdar R, Shroff N. A minimum cost heterogeneous sensor network with a lifetime constraint. *IEEE Transactions on Mobile Computing* 2005; **4**(1): 4–15.
23. Pathan A-SK, Heo G, Hong CS. A Secure Lightweight Approach of Node Membership Verification in Dense HDSN. In Proceedings of the IEEE Military Communications Conference (IEEE MILCOM'07), October 29-31, Orlando, Florida, USA, 2007, pp. 1-6.
24. Carman DW, Kruss PS, Matt BJ. Constraints and Approaches for Distributed Sensor Network Security, NAI Labs Technical Report # 00-010, NAI Labs, The Security Research Division, Glenwood, MD, USA, dated 1 September, 2000.
25. Pathan A-SK, Hong CS. SERP: secure energy-efficient routing protocol for densely deployed wireless sensor networks. *Annals of Telecommunications* 2008; **63**: (9–10): 529–541.
26. Israel AB, Greville TNE. Generalized Inverses: Theory and Applications John Wiley & Sons: New York, 1974.
27. Boullion TL, Odell PL. Generalized Inverse Matrices Wiley-Interscience: New York, 1971.
28. Haque MM, Pathan A-SK, Hong CS, Huh E-N. An asymmetric key-based security architecture for wireless sensor networks. *KSII Transactions on Internet and Information Systems* 2008; **2**(5): 265–279.
29. Rhee MY. Internet Security Cryptographic principles, algorithms and protocols John Wiley & Sons Ltd, West Sussex, England, ISBN 0-470-85285-2, pp. 165–172. 2003.
30. Gabrielli A, Mancini LV, Setia S, Jajodia S. Securing Topology Maintenance Protocols for Sensor Networks: Attacks and Countermeasures. In Proceedings of the 1st IEEE/CreateNet International Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm'05), Athens, Greece, September 2005, 101–112.
31. www.willow.co.uk/html/mpr5x0-_mica2dot_series.html [05 October], 2010.