RESEARCH ARTICLE OPEN ACCESS

An Efficient Security Mechanism for Data Reporting In Wireless Sensor Networks

P. Jeyabharathi M.E[1], A. Sivasankari M.E[2], M. Maharasi MCA,M.Phil, M.E[3]
[1,2,3]Assistant Professor, Department of MCA, Dr. Sivanthi Aditanar College of Engineering Tiruchendur, Tuticorin Dist, India

Abstract:
Wireless sensor networks consist of a large number of small sensor nodes having limited computation capacity, restricted memory space, limited power resource, and short-range radio communication device. In these scenarios, sensor networks may suffer different types of malicious attacks. The adversaries can inject false data reports via compromised nodes and launch DoS attacks against legitimate reports. Recently, a number of filtering schema for removing false data report .But due to lack of strong filtering capacity and not support dynamic sensor networks The main objective of this application is to propose a dynamic en-route filtering scheme that addresses both false report injection and DoS attacks in wireless sensor networks. In our scheme, each node has a hash chain of authentication keys used to endorse reports; meanwhile, a legitimate report should be authenticated by a certain number of nodes. First, each node release its key to forwarding nodes. Then, after sending reports, the sending nodes disclose their keys, allowing the forwarding nodes to verify their reports. The Hill Climbing key release approach is used in this approach which ensures the nodes closer to data sources have stronger filtering capacity The DoS attacks can be solved with multipath routing to deal with the topology changes of sensor networks. The proposed filtering scheme can drop false reports earlier with a lower memory requirement, especially in highly dynamic sensor networks.
Keywords-Data reporting, en-route filtering scheme, wireless sensor networks.

I. INTRODUCTION

Wireless sensor network (WSN) consists of a large number of sensor nodes, which are tiny, low-cost, low-power radio devices dedicated to performing certain functions such as collecting various environmental data and sending them to sink nodes. In military applications sensor nodes may be deployed in hostile environments such as battlefields to monitor the activities of enemy forces. In these scenarios, sensor networks may suffer different types of malicious attacks.

Recently, several schemes such as SEF [9], IHA [10], CCEF [7], LBRS [8], and LEDS [6] have been proposed to address false report injection attacks and/or DoS attacks. However, they all have some limitations. SEF is independent of network topology, but it has limited filtering capacity and cannot prevent impersonating attacks on legitimate nodes. IHA has a drawback, that is, it must periodically establish multihop pair wise keys between nodes. Moreover, it asks for a fixed path between the base station and each cluster-head to transmit messages in both directions, which cannot be guaranteed due to the dynamic topology of sensor networks or due to the use of some underlying routing protocol such as GPSR [4]. CCEF also relies on the fixed paths as IHA does and it is

even built on top of expensive public-key operations. More severely, it does not support en-route filtering. LBRS and LEDS utilize location-based keys to filter false reports. They both assume that sensor nodes can determine their locations in a short period of time. However, this is not practical, because many localization approaches [1], [3] take quite long and are also vulnerable to malicious attacks [2], [5].

In LBRS, *report disruption attacks* are simply discussed, but no concrete solution is proposed. LEDS tries to address *selective forwarding attacks* by allowing a whole cell of nodes to forward one report, however, this incurs high communication overhead. In this paper, we propose a dynamic en-route filtering scheme to address both false report injection attacks and DoS attacks in wireless sensor networks. In our scheme, sensor nodes are organized into clusters. Each legitimate report should be validated by multiple message authentication codes (MACs), which are produced by sensing nodes using their own authentication keys. The authentication keys of each node are created from a hash chain. Before sending reports, nodes disseminate their keys to forwarding nodes using *Hill Climbing* approach. Then, they send reports in rounds. In each round, every sensing node endorses its reports using a new key and then discloses the key to forwarding nodes. Using the disseminated and disclosed keys, the forwarding nodes can validate the reports. In our scheme, each node can monitor its neighbors by overhearing their broadcast, which prevents the compromised nodes from changing the reports. Report

forwarding and key disclosure are repeatedly executed by each forwarding node at every hop, until the reports are dropped or delivered to the base station.
Our scheme has two advantages:

- We design the *Hill Climbing* approach for key dissemination, which ensures that the nodes closer to clusters hold more authentication keys than those closer to the base station do. This approach not only balances memory requirement among nodes, but also makes false reports dropped as early as possible.
- Multipath routing is adopted when disseminating keys to forwarding nodes, which not only reduces the cost for updating keys in highly dynamic sensor networks, but also mitigates the impact of selective forwarding attacks.

Simulation results show that, compared to existing ones, our scheme can drop false reports earlier with a lower memory requirement, especially in the networks whose topologies change frequently.

II. RELATED WORK

Existing Schemes for Filtering False Reports

A statistical en-route filtering (SEF) scheme [9] based on probabilistic key distribution. In SEF, a global key pool is divided into *n* partitions, each containing *m* keys. Every node randomly picks *k* keys from one partition. When some event occurs, each sensing node creates a MAC for its report using one of its random keys. The cluster-head aggregates the reports from the sensing nodes and guarantees each aggregated report contains *T* MACs that are generated using the keys from different *T* partitions, where *T* is a predefined security parameter. Given that no more than *T-1* nodes can be compromised, each forwarding node can detect a false report with a probability proportional to ***1/n***. In addition, since the keys are shared by multiple nodes, the compromised nodes can impersonate other nodes and report some forged events that "occur" in other clusters.
An interleaved hop-by-hop authentication (IHA) scheme [10]. In this scheme, the base station periodically initiates an association process enabling each node to establish pairwise keys with other nodes that are *t+1* hops away, where *t* is a security threshold. In IHA, each sensing node generates a MAC using one of its multihop pairwise keys, and a legitimate report should contain *t+1* distinct MACs. Since each multihop pairwise key is distinct, IHA can tolerate up to compromised nodes in each cluster instead of in the whole network as SEF does. However, IHA requires the existence of a fixed path for transmitting control messages between the base station and every cluster-head, which cannot be guaranteed by some routing protocols such as GPSR [4] .

A commutative cipher based en-route filtering (CCEF) scheme [9]. In CCEF, each node is reloaded with a distinct authentication key. When a report is needed, the base station sends a session key to the cluster-head and a witness key to every forwarding node along the path from itself to the cluster-head. The report is appended with multiple MACs generated by sensing nodes and the cluster-head. When the report is delivered to the base station along the same path, each forwarding node can verify the cluster-head's MAC using the witness key. The MACs generated by sensing nodes can be verified by the base station only. CCEF has several drawbacks. First, it relies on fixed paths as IHA does. Second, it needs expensive public-key operations to implement commutative ciphers. Third, it can only filter the false reports generated by a malicious node without the session key instead of those generated by a compromised cluster-head or other sensing nodes.

A location-based resilient security(LBRS) solution [8]. In LBRS, a sensing field is divided into square cells, and each cell is associated with some cell keys that are determined based on the cell's location. Each node stores two types of cell keys. One type contains the keys bounded to their sensing cells to authenticate the reports from those cells. A location-aware end-to-end data security (LEDS) scheme that can address false report injection and some DoS attacks. Like LBRS, LEDS assumes that sensor nodes can generate the location-based keys bounded to cells within a secure short time slot. LEDS provides end-to-end security by allowing sensing nodes to encrypt their messages using the cell keys. A legitimate report contains *T* distinct shares produced from the encrypted message using nodes' secret keys, where the base station can always recover the original message from any *t(t<T)* valid shares. In addition, LEDS addresses selective forwarding attacks by letting the whole cell of nodes to forward reports, which incurs high communication overhead.

III. PROBLEM STATEMENT

A. System Model

We model the communication region of wireless sensor nodes as a circle area of radius *r*, which is called the *transmission range*. We only consider the bidirectional links between neighbor nodes and assume that sensor nodes simply discard or ignore those links that are not bidirectional. Based on these assumptions, we say that two nodes must be the neighbor of each other and can always communicate with each other if the distance between them is no more than *r*.

Wireless sensor nodes may be deployed into some target field to detect the events occurring within the field. For example, in a military application, they may be deployed to a battlefield to detect the activities of enemy forces. We assume that sensor nodes form a number of clusters after deployment, each containing at least *n* nodes. In each cluster, one node is randomly selected as the *cluster-head*. To balance energy consumption, all nodes within a cluster take turns to serve as the cluster-head. That means physically there is no difference between a cluster-head and a normal

node because the cluster-head performs the same sensing job as the normal node.

Fig. 1 illustrates the organization of sensing nodes in wireless sensor networks. In the figure *CH* and *BS* denote *Cluster-Head* and *Base Station* respectively. *u1~u5* are forwarding are forwarding nodes, and *v1~v8* are sensing nodes (they can also serve as the forwarding nodes for other clusters). The black dots represent the compromised nodes, which are located either in the clusters or en-route.

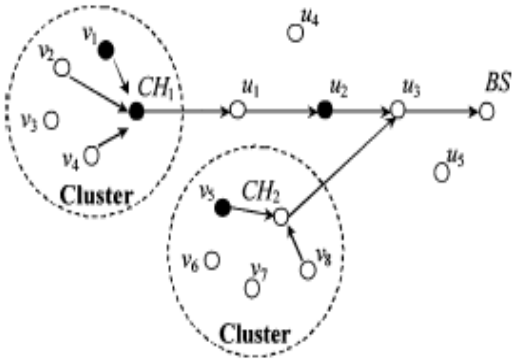

Fig. 1. Sensor nodes are organized into clusters. The big dashed circles outline the regions of clusters. CH and BS denote *Cluster-Head* and *Base Station* respectively. u_1~u_5 are forwarding nodes, and v_1~v_8 are sensing nodes (they can also serve as forwarding nodes for other clusters). The black dots represent the compromised nodes, which are located either within the clusters or en-route.

IV. OUR SCHEME

A. Overview

When an event occurs within some cluster, the cluster-head collects the *sensing reports* from sensing nodes and aggregates them into the *aggregated reports*. Then, it forwards the aggregated reports to the base station through forwarding nodes. In our scheme, each sensing report contains one MAC that is produced by a sensing node using its authentication key (called *auth-key* for short), while each aggregated report *t* contains distinct MACs, where *t* is the maximum number of compromised nodes allowed in each cluster.

In our scheme, each node possesses a sequence of auth-keys that form a hash chain. Before sending the reports, the cluster-head disseminates the first auth-keys of all nodes to the forwarding nodes that are located on multiple paths from the cluster-head to the base station. The reports are organized into rounds, each containing a fixed number of reports. In every round, each sensing node chooses a new auth-key to authenticate its reports. To facilitate verification of the forwarding nodes, the sensing nodes disclose their auth-keys at the end of each round. Meanwhile, to prevent the forwarding nodes from abusing the disclosed keys, a forwarding node can receive the disclosed auth-keys, only after its upstream node overhears that it has already broadcast the reports. Receiving the disclosed keys, each forwarding node verifies the reports, and informs its next-hop node to forward or drop the reports based on the verification result. If the reports are valid, it discloses the keys to its next-hop node after overhearing. The processes of verification, overhearing, and key disclosure are repeated by the forwarding nodes at every hop until the reports are dropped or delivered to the base station.

Specifically, our scheme can be divided into three phases: *key predistribution phase*, *key dissemination phase*, and *report forwarding phase*. In the *key redistribution phase*, each node is preloaded with a distinct seed key from which it can generate a hash chain of its auth-keys. In the *key dissemination phase*, the cluster-head disseminates each node's first auth-key to the forwarding nodes, which will be able to filter false reports later. In the *report forwarding phase*, each forwarding node verifies the reports using the disclosed auth-keys and disseminated ones. If the reports are valid, the forwarding node discloses the auth-keys to its next-hop node after overhearing that node's broadcast. Otherwise, it informs the next-hop node to drop the invalid reports. This process is repeated by every forwarding node until the reports are dropped or delivered to the base station.

Fig. 2 demonstrates the relationship between the three phases of our scheme. *Key predistribution* is performed before the nodes are deployed, e.g., it can be done offline. *Key dissemination* happens before the sensing nodes begin to send the reports. It may be executed periodically depending on how often the topology is changed. Every time the latest (unused) auth-key of sensing nodes will be disseminated. *Report forwarding* occurs at each forwarding node in every round.

A. Detailed Procedure

In the section, we discuss the procedure of each phase in detail.

1) Key Predistribution Phase*:* Key predistribution

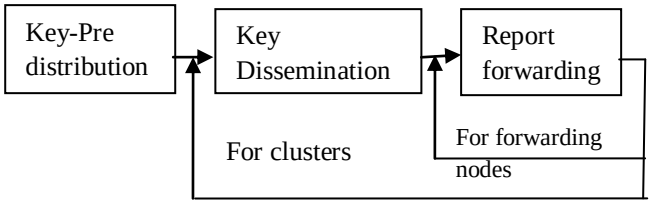

Fig. 2. The relationship between three phases of our scheme. Key predistribution is preformed only once. Key dissemination is executed by clusters periodically. Report forwarding happens at each forwarding node in every round.

needs to be performed only once. It consists of two steps.

Step1: Each node is preloaded with a distinct seed key. From the seed key, it can generate a sequence of auth-keys using a common hash function *h* . Thus, each node's authkeys form a hash chain. Let m denote

the length of hash chain. Given node v_i as well as its seed key $k_m^{v_i}$, its auth keys can be calculated as follows:

$$k_{m-1}^{v_i} = h\left(k_m^{v_i}\right)$$
$$k_{m-2}^{v_i} = h\left(k_{m-1}^{v_i}\right)$$
$$\vdots$$
$$k_1^{v_i} = h^{m-1}\left(k_m^{v_i}\right)$$

Besides the seed key, each node is also equipped With $l+1$ secret keys, where l keys (called y -keys) are randomly picked from a global key pool (called y -key pool) of size v , and the rest (called z -key) is randomly chosen from another global key pool (z -key pool) of size w . Among n nodes of a cluster, we assume that there are at least t nodes each having a distinct z -key.

2) Key Dissemination Phase*:* In our scheme, the cluster-head discloses the sensing nodes' auth-keys after sending the reports of each round. However, it is vulnerable to such an attack that a malicious node can pretend to be a cluster-head and inject arbitrary reports followed by falsified auth-keys. To prevent this attack, we enforce *key dissemination*, that is, the cluster-head should disseminate the *first* auth-keys of all nodes to the forwarding nodes before sending the reports in the first round that can be seen in fig3. By using the disseminated keys, the forwarding nodes can verify the authenticity of the disclosed auth-keys, which are in turn used to check the validity and integrity of the reports.

Key dissemination should be performed periodically in case that some forwarding nodes aware of the disseminated keys become failed, especially when the network topology is highly dynamic. In this case (of redissemination), the *first unused*, instead of the *first*, auth-keys will be disseminated. The first unused auth-key of a node is called the *current auth-key* of that node. When none of a node's auth-keys has ever been used, the current auth-key is just the first auth-key of its hash chain.The detailed procedure of key dissemination phase is as follows

Fig. 3. Selecting the cluster head and forwarding node

Step1: Each node constructs an *Auth* message, which contains $l+1$copies of its current auth-key, each encrypted using a different one of its secret keys.
Step2: The cluster-head collects the *Auth* messages from all nodes and aggregates them into message *k(n)*

$$k(n) = \langle Auth(v_{i)}, \ldots \ Auth(v_{n)} \rangle$$

where $v_1 \ldots v_n$ are the nodes of the cluster.
Step3: The cluster-head chooses forwarding nodes from its neighbors and forwards them a message k(n).
Step4: When a forwarding node receives , it performs the following operations:
1) It verifies k(n) to see if k(n) contains at least t distinct indexes of z-keys. If not, this k(n) is assumed to be forged and should be dropped.
2) It checks the indexes of secret keys in k(n) to see if it has any shared key. When a shared secret key is found ,it decrypts the corresponding auth-key using that key and stores the auth-key in its memory. Obviously, it must assure that the decryption key is the correct one by checking the index encrypted along with the authkey. Otherwise, it discards k(n).
3) k(n) does not need to be disseminated to the base station. We define h_{max} as the maximum number of hops that k(n) should be disseminated. Each forwarding node discards the k(n) that has already been disseminated h_{max} hops. Otherwise, it forwards k(n) to other q downstream neighbor nodes, which are selected using the same metric as the cluster-head uses. Each node receiving k(n) repeats these operations, until k(n) gets to the base station or has been disseminated h_{max} hops.
3) Hill Climbing: *Hill Climbing* involves two variations, one for the key predistribution phase and the other for the key dissemination phase.
The first variation is: In Step2 of the key distribution phase, instead of picking y-keys from a global key pool, each node selects each of its y-keys randomly from an independent hash chain. Specifically, the original y-key pool is partitioned into l equal-sized hash chains, each containing $\frac{v}{l}$ keys that are generated from a distinct seed key. It is easy to know that a forwarding node holding a larger index y-key can always decrypt a sensing node's auth-key from k(n) as long as the sensing node's y -key has a smaller index. Inspired by this, we propose the second variation. That is, in Step4 of the key dissemination phase, after a forwarding node decrypts an auth-key from k(n) , it updates k(n) by encrypting the auth-key using its own y-key and then forwards the updated to its downstream neighbor nodes.
4) Report Forwarding Phase*:* In this phase, sensing nodes generate sensing reports in rounds. Each round contains a fixed number of reports, e.g., 10 reports, where this number is predetermined before nodes are deployed. In each round, every sensing node chooses

a new auth-key, i.e., the node's current auth-key, to authenticate its reports.
Given node , its sensing report is
$R(v_i)=\{E,v_i,j_i,MAC(E,k_{j_i}^{v_i}\}$

where denotes the event information, j_i is the index of v_i's current auth-key, and is theMAC(E, $k_{j_i}^{v_i}$) generated from E using key $k_{j_i}^{v_i}$. In each round, the cluster-head generates the aggregated reports and forwards them to next hop, i.e., one of its q selected downstream forwarding nodes. Then, it discloses the sensing nodes' auth-keys after overhearing the broadcast from the next-hop node. The reports are forwarded hop-by-hop to the base station. At every hop, a forwarding node verifies the validity of reports using the disclosed keys and informs its own next-hop node the verification result. The same procedure is repeated at each forwarding node until the reports are dropped or delivered to the base station.

V. SIMULATION RESULTS

1)Fraction of False Reports Filtered Versus Number of Hops They Traveled:

We first consider the case that *t-1* compromised nodes are within the same cluster. We assume a static environment in which all nodes are in ON state. Fig. 4 illustrates how the fraction of false reports filtered increases as the number of hops that they traveled grows. In our scheme, *K(n)* is disseminated within at most h_{max}=10 hops and each node stores at most mem=50 keys.

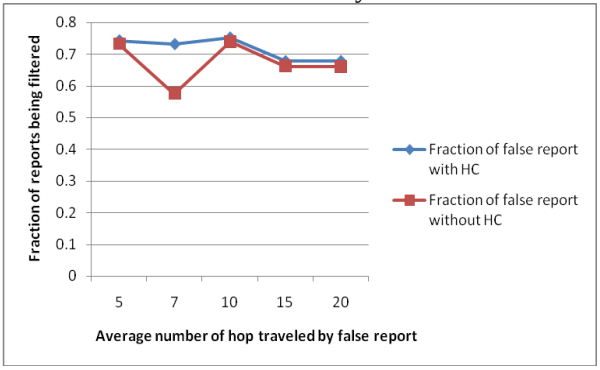

Fig. 4. The fraction of false reports filtered as a function of the number of hops that they traveled (q=2 for our scheme).

2)Filtering Capacity Versus Maximum Number of Hops for Key Dissemination:

Fig. 5 shows the impact of h_{max} on the filter capacity of our scheme. Typically, disseminating auth-keys farther makes more nodes capable of filtering false reports. At the same time, the limited memory size forces each node to discard more auth-keys of each cluster in order to accommodate more clusters. Hence, increasing the value of h_{max} is not always helpful. Fig. 5 indicates that the best value of h_{max} is between 5 to 10 because 90% of false reports have been dropped within
10 hops, as shown in Fig.4.

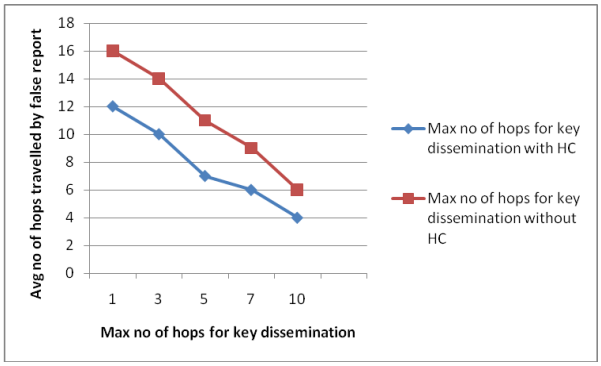

Fig. 5. The average number of hops traveled by false reports as a function of the maximum number of hops for key dissemination. (q=2 for our scheme).

VI. CONCLUSION

In this paper, we propose a dynamic en-route quarantine scheme for filtering false data injection attacks and DoS attacks in wireless sensor networks. In our scheme, each node uses its own auth-keys to authenticate their reports and a legitimate report should be endorsed by nodes. The auth-keys of each node form a hash chain and are updated in each round. The cluster-head disseminates the first auth-key of every node to forwarding nodes and then sends the reports followed by disclosed auth-keys. The forwarding nodes verify the authenticity of the disclosed keys by hashing the disseminated keys and then check the integrity and validity of the reports using the disclosed keys. According to the verification results, they inform the next-hop nodes to either drop or keep on forwarding the reports. This process is repeated by each forwarding node at every hop. Our scheme has several advantages: 1) Compared with others, our scheme can drop false reports much earlier even with a smaller size of memory. 2) The uncompromised nodes will not be impersonated because each node has its own auth-keys. Therefore, once the compromised nodes are detected, the infected clusters can be easily quarantined. 3) Our *Hill Climbing* key dissemination approach increases filtering capacity greatly and balances the memory requirement among nodes. 4) Each node has multiple downstream nodes that possess the necessary key information and are capable of filtering false reports. This not only makes our scheme adaptive to highly dynamic networks, but also mitigates the impact of selective forwarding attacks. 5) Monitored by its upstream nodes and neighbors, the compromised nodes have no way to contaminate legitimate reports or generate false control messages. However, to achieve these advantages we have to make some tradeoffs: 1) Our scheme is more complicated than SEF by introducing extra control messages such ask(n) ,k(t) and OK . 2) The introducing of extra control messages triples the delay of reports. 3) Our scheme requires each node to monitor its downstream nodes and neighbors, which can be achieved by using only bidirectional links. Therefore, sensor nodes have

to discard all directed links. 4) In our scheme, each node uses the same auth-key to authenticate all of its reports in the same round. Therefore, this auth-key can only be disclosed after the forwarding nodes forward the reports to their next-hop nodes, which increases memory overhead of the forwarding nodes. 5) Our scheme can not be easily coordinated with other energy-efficient protocols, because in our scheme each node has to be awake until it overhears the broadcast of its next-hop
node. Further work includes how to take advantage in our scheme of various energy-efficient data aggregation and dissemination protocols for wireless sensor networks.

REFERENCES

[1] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low cost outdoor localization for very small devices," *IEEE Personal Commun. Mag.*,vol. 7, no. 5, pp. 28–34, Oct. 2000.

[2] S. Capkun and J. Hubaux, "Secure positioning of wireless devices with application to sensor networks," in *Proc. IEEE INFOCOM*, 2005, vol. 3, pp. 1917–1928

[3] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher, "Rangefree localization schemes in large scale sensor network," in *Proc. ACM MobiCom*, 2003, pp. 81–95.

[4] B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. ACM MobiCom*, 2000, pp. 243–254.

[5] L. Lazos and R. Poovendran, "SeRLoc: Secure range-independent localization forWireless sensor networks," in *Proc. ACMWiSe*, 2004, pp. 21–30

[6] K. Ren, W. Lou, and Y. Zhang, "LEDS: Providing location-aware end-to-end data security in wireless sensor networks," in *Proc. IEEE INFOCOM*, 2006, pp. 1–12.

[7] H. Yang and S. Lu, "Commutative cipher based en-route filtering in wireless sensor networks," in *Proc. IEEE VTC*, 2004, vol. 2, pp.1223–1227.

[8] H. Yang, F. Ye, Y. Yuan, S. Lu, and W. Arbaugh, "Toward resilient security in wireless sensor networks," in *Proc. ACM MobiHoc*, 2005, pp. 34–45

[9] F. Ye, H. Luo, S. Lu, and L. Zhang, "Statistical en-route detection and filtering of injected false data in sensor networks," in *Proc. IEEE INFOCOM*, 2004, vol. 4, pp. 2446–2457.

[10] S. Zhu, S. Setia, S. Jajodia, and P. Ning, "An interleaved hop-by-Hop authentication scheme for filtering of injected false data in sensor networks," in *Proc. IEEE Symp. Security Privacy*, 2004, pp. 259–271

ABOUT THE AUTHORS

P.Jeyabharathi She is presently working as a Assistant Professor in Dr.Sivanthi Aditanar College of Engineering, Tiruchendur. She has done her M.E (CSE) in Francis Xavier Engineering College, Anna University @ Tirunelveli in 2011. She received her B.E degree from Dr.Sivanthi Aditanar College of Engineering. Anna University @ Chennai in 2009. She had presented papers in national and International Conferences.

A.Sivasankari She is presently working as a Assistant Professor in Dr.Sivanthi Aditanar College of Engineering, Tiruchendur. She has done her M.E (CSE) in Pavendhar Bharathidasan college of Engg. & Technology, Anna University @ Trichy in 2011. She received her B.E degree from Dr.Sivanthi Aditanar College of Engineering. Anna University @ Chennai in 2009. She had presented papers in national and International Conferences.

M.Maharasi She is presently working as a Assistant Professor in Dr.Sivanthi Aditanar College of Engineering, Tiruchendur. She has done her M.E (CSE) in Dr.Sivanthi Aditanar College of Engineering, Anna University @ Tiruchendur in 2010. She received her M.Phil degree from Manonmaniam Sundaranar University at Tirunelveli in 2004. She received her MCA degree in Sri Saratha College for women @ Bharathidasan University, Karur in 1996. She received her B.sc (computer Science) degree in Cauvery College for women Bharathidasan University @ Trichy in 1993. She has 12 years of teaching experience in this field. She had presented papers in national and International Conferences.