Abstract
Intruders can inject bogus reports via compromised nodes and launch DoS attack against legitimate reports in wireless sensor networks (WSNs). For many applications in wireless sensor networks, users may want to continuously extract data from the networks for analysis later. In this paper, energy efficient Sleep/awake scheduling algorithm is proposed along with the dynamic en-route filtering scheme. As the sensor nodes are allowed to sleep periodically under certain condition, it reduces the energy consumption of all nodes including cluster head. The cluster head need no of collecting all the sensor nodes data instead they need only the cluster member that are awake. The dynamic en-route filtering scheme addresses both false report injection and DoS attacks in wireless sensor networks. Each node send its key to forwarding nodes and then disclose their keys by verify their reports using forwarding nodes. In Hill Climbing, key dissemination approach ensures the nodes closer to data sources has stronger filtering capacity and also drops fabricated reports en-route without symmetric key sharing. Thus this approach is used to achieve stronger security protection. The cluster head is required to collect the data from all the sensor node, which makes it overloaded all the time hence Sleep/awake scheduling algorithm is used which will avoid this problem.

Keywords
Clustering, Data Reporting, En-Routte Filtering Scheme, Hill Climbing Approach, Sleep/Awake Scheduling Algorithm

I. Introduction
Wireless Sensor Networks (WSNs) have a broad range of applications, such as battlefield surveillance, environmental monitoring, and disaster relief. A sensor network consists of a set of autonomous sensor nodes which spontaneously create communication links, and then, collectively perform tasks without help from any central servers. Also 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 military applications, sensor nodes may be deployed in hostile environments such as battlefields to monitor the activities of enemy forces.

In this scenario, 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 [18]. This approach mainly focus on routing security in wireless sensor networks. 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. Sensor networks may suffer different types of malicious attacks. One type is called false report injection attacks [24], in which adversaries inject into sensor networks the false data reports containing nonexistent events or faked readings from compromised nodes. These attacks not only cause false alarms at the base station, but also drain out the limited energy of forwarding nodes.

In this 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. Cluster head collect all the data from sensing node so cluster head has much burden, so Sleep/awake Scheduling Algorithm [3, 9] was proposed, which is better than Statistical En-Routte Filtering (SEF) [14]. It’s a critical design is reducing the power consumption. It’s a practical algorithm for data aggregation that collects the overall distribution of the sensor data.

Proposed scheme has following three advantages:

• Hill Climbing Dissemination, 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 forwarding nodes, which not only reduces the cost for updating symmetric keys in highly dynamic sensor networks, but also mitigates the impact of selective forwarding attacks.

• Sleep/Awake Scheduling Algorithm will reduce high energy consumption in wireless sensor network.

II. Related Work
Existing filtering schemes is first discussed here, and then going to discuss about application specific protocol. The routing strategies of these protocols affect the way that sensor nodes can exchange and disseminate key information.

A. Existing Schemes for Filtering False Reports
Yang et al. presented a Commutative Cipher Based En-Routte Filtering (CCEF) scheme [8]. In CCEF, each node is preloaded 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. 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 [6]. CCEF has several drawbacks. First, it relies on fixed paths. 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.

Recently, Ren et al. proposed a Location-aware End-to-End Data Security (LEDSS) [10], scheme that can address false report injection and some DoS attacks. It assumes that sensor nodes can generate the location-based keys and cells within a secure short time slot. LEDSS 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. LEDSS addresses selective forwarding attacks by letting the whole cell of nodes to forward reports, which inures high communication overhead. It is also vulnerable to malicious attacks [4]. In LEDSS, the reports are forwarded through cells along report-auth routes. Each node stores the authentication keys shared between its cell and others in its downstream report-auth area and on the report-auth route.

B. Application-Specific Protocol

Wendi et al. proposed a low-energy adaptive clustering hierarchy (LEACH) [1], a protocol architecture for micro sensor networks that combines the ideas of energy-efficient cluster-based routing and media access together with application-specific data aggregation to achieve good performance in terms of system lifetime, latency, and application-perceived quality. LEACH includes a new distributed cluster formation technique that enables self-organization of large numbers of nodes, algorithms for adapting clusters and rotating cluster head positions to evenly distribute the energy load among all the nodes, and techniques to enable distributed signal processing to save communication resources. LEACH can improve system lifetime by an order of magnitude compared with general-purpose multipath approaches.

In LEACH, the nodes organize themselves into local clusters, with one node acting as the cluster head. All non-cluster head nodes transmit their data to the cluster head, while the cluster head node receives data from all the cluster members, performs signal processing functions on the data (e.g., data aggregation), and transmits data to the remote BS. Therefore, being a cluster head node is much more energy intensive than being a non-cluster head node. If the cluster heads were chosen a priori and fixed, the cluster head is responsible for aggregating these sensing reports into the aggregated reports and forwarding these aggregated reports to the base station through some forwarding nodes. However, 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. 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. In the figure, CH and BS denotes the Cluster-Head, Base Station respectively. u1 – u5 are forwarding nodes, and v1 – v8 are sensing nodes. The black dots represent the compromised nodes, which are located either in the clusters or en-route.

C. Routing Protocols of Sensor Network

Braginsky et al. proposed Rumor [1] [5] routing protocol. In Rumor, when a sensing node detects some event, it creates an agent that is actually a message containing the routing information about the event. The agent follows a straight path to leave from the sensing node and is associated with a maximum TTL. Each node passed by the agent learns the route to the event. If the base station is interested in some event, it sends out a query message. The movement pattern of a query message is similar to that of an agent. When a query message is delivered to a node who knows the route to the event, a path between the base station and the sensing node (the event) can be established. Although our scheme take advantage of any routing protocol that is designed for wireless sensor networks.

III. Problem Statement

A. System Model

The communication region of wireless sensor nodes is modeled as a circle area of radius, which is called the transmission range. The bidirectional links is considered between neighbor nodes and assume that sensor nodes simply discard or ignore those links that are not bidirectional. Based on these assumptions, the two nodes must be the neighbor of each other and can always communicate with each other. Wireless sensor nodes may be deployed into some target field to detect the events occurring within the field. These nodes detecting the event are called sensing nodes. They generate and broadcast the sensing reports to the cluster-head. The cluster-head is responsible for aggregating these sensing reports into the aggregated reports and forwarding these aggregated reports to the base station through some forwarding 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 denotes the Cluster-Head, Base Station respectively. u1 – u5 are forwarding nodes, and v1 – v8 are sensing nodes. The black dots represent the compromised nodes, which are located either in the clusters or en-route.

B. Our Goal

The goal is to improve the energy efficiency by using Sleep/awake algorithm and it reduce the Cluster head burden. In Sleep/awake scheduling algorithm, when a cluster member is awake, the cluster head checks if the member’s data values are within the error bound with high probability. If yes, the cluster head will send a message to power off the member. If no, the cluster head will send a message to awake the member. This also increases the network life time. It can also offer stronger filtering capacity and drop false reports earlier with the acceptable memory requirement.
where the filtering capacity is defined as an average number of hops that a false report can travel. It can address or mitigate an impact of DoS attacks such as selective forwarding attacks and report disruption attacks. It should not rely on any fixed paths between a base station and cluster-heads to transmit messages. It should prevent uncompromised nodes from being impersonated. Therefore, when the compromised nodes are detected, infected clusters can be easily quarantined by the base station.

Compared to existing ones, this scheme is expected to achieve the following goals:
1. It can offer stronger filtering capacity and drop false reports earlier with an acceptable memory requirement, where the filtering capacity is defined as the average number of hops that a false report can travel.
2. It reduces the Cluster head burden by avoiding the unwanted data collection.
3. It should not rely on any fixed paths between the base station and cluster-heads to transmit messages.
4. It should prevent the uncompromised nodes from being impersonated. Therefore, when the compromised nodes are detected, the infected clusters can be easily quarantined by the base station.

IV. Our Scheme

A. Sleep/Awake Scheduling Algorithm

When a cluster member is awake, the cluster head checks if the member’s data values are within the error bound with high probability. If yes, the cluster head will send a message to power off the member. The condition should be the confidence level \( \alpha_m \) is higher than the threshold \( \alpha_{\text{threshold}} \), i.e.,

\[
\text{condition does not hold.}
\]

The main advantage of the sleep/awake scheduling algorithm is the cluster node doesn’t want to collect all the data of the sensing node. The sensing nodes that wish to send the data alone is awake rest will go to sleep state. So the Cluster head burden is reduced. Aggregation will also improve the network life time. Hence the sleep/awake scheduling will also save energy. Compared to Sleep/awake scheduling SEF is independent of network topology, but it has limited filtering capacity and cannot prevent impersonating attacks on legitimate nodes.

B. Elimination of False Report Injection

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 this 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, in this scheme it can be divided into three phases as shown in fig. 3: (a) Key pre-distribution phase, (b) Key dissemination phase, (c) Report forwarding phase. In the key pre-distribution 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 discloses 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. The detailed procedures of three phases are discussed here. In the key pre-distribution phase, each node is preloaded with secret keys $k_1$,...,$k_m$ from the seed key $km$. In the key dissemination phase, the cluster-head disseminates the auth-keys of all nodes by message $K(n)$ to $q$ downstream neighbor nodes. Every downstream node decrypts some auth-keys from $K(n)$, and further forwards $K(n)$ to $q$ more downstream neighbor nodes, which then repeat the same operation. In the report forwarding phase, each forwarding node en-route performs the following steps:

1. It receives the reports from its upstream node.
2. If it receives confirmation message OK, then forwards the reports to its next-hop node. Otherwise, it discards the reports.
3. It receives the disclosed auth-keys within message $K(n)$ and verifies the reports by using the disclosed keys.
4. It informs its next-hop node the verification result.

### 1. Key Dissemination Phase

Here the cluster-head discloses the sensing nodes’ auth-keys after sending the reports of each round. However, the cluster-head should disseminate the first auth-keys of all nodes to the forwarding nodes before sending the reports in the first round. 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 re-dissemination), 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 cluster-head collects the Auth messages from all nodes and aggregates them into message $K(n)$.

$$K(n)=\{\text{auth}(v_1),...\text{auth}(v_n)\}$$

Where $v_1$,...,$v_n$ are the nodes of the cluster. The cluster-head chooses $q(\geq 1)$ forwarding nodes from its neighbors and forwards them a message, $K(n)$. These $q$ nodes can be selected based on different metrics such as the distance to the base station, the link quality, the amount of energy available, the speed of energy consumption, or a combination of all.

When a forwarding node receives $K(n)$, it performs the following operations:

1. It verifies $K(n)$ to see if $K(n)$ contains at least distinct indexes of -keys. If not, this $K(n)$ is assumed to be forged and should be dropped. 2) $K(n)$ does not need to be disseminated to the base station. Here $\text{hmax}$ is defined as the maximum number of hops that $K(n)$ should be disseminated. Each forwarding node discards the $K(n)$ that has already been disseminated $\text{hmax}$ hops. Otherwise, it forwards $K(n)$ to other 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.

### 2. Hill Climbing

Here two important observations are introduced. First, when multiple clusters disseminate keys at the same time, some forwarding nodes need to store the auth-keys of different clusters. First is that the nodes closer to the base station need to store more auth-keys than others (typically those closer to clusters) do because they are usually the hot spots and have to serve more clusters. Second, the false reports are mainly filtered by the nodes closer to clusters, while most nodes closer to the base station have no chance to use the auth-keys they stored for filtering.

### 3. Report Forwarding Phase

In this phase, sensing nodes generate sensing reports in rounds. In each round, every sensing node chooses a new auth-key, i.e., the node’s current auth-key, to authenticate its reports. In each round, the cluster-head generates the aggregated reports and forwards them to the 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 verifies the 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 Result

For the simulation of the proposed scheme NS-2 is being used. The routing protocol adopted in our simulation is AODV (Ad hoc On demand Distance Vector). AODV as routing protocol is preferred because it does not need any central administrative system to control the routing process. Generally reactive routing protocols like AODV tend to reduce the control message overheads at the cost of increased latency in finding new routes and also it reacts relatively fast to the topology changes in the network and updates only the nodes affected by these changes. It also saves storage place and energy. The simulation consists of 40 sensor nodes and cluster heads is chosen. Out of those 40 nodes some nodes are configured to be malicious and their intention was set to drop the packets there by launching false data injection attack and Dos attacks. In this scheme offers each node a higher detecting probability.
This scheme has several following advantages:

1. A dynamic en-route monitored by its upstream nodes and neighbors, the compromised nodes have no way to contaminate legitimate reports or generate false control messages.
2. The performance evaluation and security analysis show that our key management scheme for sensor nodes. The performance evaluation and security analysis show that our key management scheme for sensor nodes.
3. Hill Climbing Dissemination key approach increases filtering capacity greatly and balances the memory requirement among nodes.
4. The unimpersonated nodes will not be impersonated because an each node has its own auth-keys. Therefore, once the compromised nodes are detected, the infected clusters can be easily quarantined.
5. An 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.

VI. Conclusion

We propose a novel routing-driven key management scheme, which only establishes shared keys for neighbor sensors that communicate with each other. We utilize Elliptic curve cryptography (ECC) cryptography in the design of an efficient key management scheme for sensor nodes. The performance evaluation and security analysis show that our key management scheme can provide better security with significant reductions on communication overhead, storage space and energy consumption other than other key management schemes.

This scheme has several following advantages:

1. The unimpersonated nodes will not be impersonated because an each node has its own auth-keys. Therefore, once the compromised nodes are detected, the infected clusters can be easily quarantined.
2. Hill Climbing Dissemination key approach increases filtering capacity greatly and balances the memory requirement among nodes.
3. An each node has multiple downstream nodes that possess

References


Fig. 4: Cluster Formation

Collect the small sensor nodes and form a cluster. Here there are 12 clusters are formed as shown in fig. 4 : 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. Using different metrics such as the distance to the base station, link quality, the amount of energy available, speed of energy consumption etc the forwarding node is selected and the message at last reach the base station. When the simulation is started the route discovery process of AODV is done and report forwarding nodes are chosen. Now the environment is ready for the sensor nodes to sense the events and report them to their respective clusters. As the simulation time progresses the malicious nodes activity starts to increase and as a result they drop the packets selectively as in case of selective forwarding attack or completely dropped by cluster head. Then the implementation of sleep/awake scheduling in which, the cluster member is awake if and only if £<0 i.e. power level must be high otherwise member will go to sleep state. At last reports are delivered to the base station i.e. the messages are delivered. By analyzing the battery power consumption we detect how much energy it has utilized.


Mrs. S. Ratalu is a student of Aditya Engineering College, Surampalem, E.G. Dist., AP, India. Presently, she is pursuing her M.Tech in this college.

Ms. T. Sudha Rani is a Sr. Assistant Professor Department of Computer Science and Engineering, Aditya Engineering college, Surampalem, E.G.Dist, AP, India.