SVE: Security using Virtual Energy for Wireless Sensor Networks

K. Naga Krishnaja and Dr. M H M Krishna Prasad
Dept. of Information Technology, UCEV, Vizianagaram, Andhra Pradesh, India

Abstract: Today most of the applications such as military operations, medical systems, aerospace systems, robotic systems, process control, factory automation, building and environmental control and smart spaces need the integration of the sensor-based systems; hence there is a need to provide security for these sensor-based applications for the healthy operations. It is very important to provide authentic and accurate data to surrounding sensor nodes. For this every node should be able to verify the authenticity and integrity of the incoming packets using the predicted value of the key generated based on the energy of the node in-order to reduce the energy consumption of the sensor nodes during the transmission. Therefore, protocols should be resilient against false data injected, and also be able to reduce the cost as the resources are limited. With the use of virtual energy of the nodes as the key to the encryption procedure the performance was improved by 60-100% of the other comparable schemes.

Keywords: Cost-efficient, Sensor nodes, Virtual-energy

1. Introduction

WSN technology will be used in a variety of application areas such as environmental, military, and commercial enterprises. For example, sensor nodes forming a network under water could be used for oceanographic data collection, pollution monitoring, assisted navigation, military surveillance, and mine reconnaissance operations.

Protocols should be resilient against false data injected into the network by malicious nodes. Securing sensor networks poses unique challenges to protocol builders because these tiny wireless devices are deployed in large numbers, usually in unattended environments, and are severely limited in their capabilities and resources (e.g., power, computational capacity, and memory). For instance, [4] a typical sensor operates at the frequency of 2.4 GHz, has a data rate of 250 Kbps, 128 KB of program flash memory, 512 KB of memory for measurements, transmit power between 100 micro W and 1 m W, and a communications range of 30 to 100 m. Therefore, protocol builders must be cautious about utilizing the limited resources onboard the sensors efficiently.

One way to eliminate injected malicious data from WSNs is to utilize an en-route filtering scheme. The en-route filtering schemes [1, 2, 3] generally utilizes two fundamental keys; static and dynamic. In static key management schemes [5], key management functions (i.e., key generation and distribution) are handled statically. That is, the sensors have a fixed number of keys loaded either prior to or shortly after network deployment. On the other hand, dynamic key management schemes [6] perform keying functions (rekeying) either periodically or on demand as needed by the network. The sensors dynamically exchange keys to communicate. Although dynamic schemes are more attack resilient than static ones, one significant disadvantage is that they increase the communication overhead due to keys being refreshed or redistributed from time to time in the network. There are many reasons for key refreshment, including: updating keys after a key revocation has occurred, refreshing the key such that it does not become stale, or changing keys due to dynamic changes in the topology.

* Corresponding author. Tel.: + 0866-2438808
E-mail address: nagakrishnaja@gmail.com.
The main motivation behind this paper is to reduce the communication cost, as it is the most dominant factor in a sensor’s energy consumption. Dynamic keying schemes go through the phase of rekeying either periodically or on demand as needed by the network to refresh the security of the system. With rekeying, the sensors dynamically exchange keys that are used for securing the communication. Hence, the energy cost function for the keying process from a source sensor to the sink while sending a message on a particular path with dynamic key-based schemes can be written as follows (assuming computation cost, $E_{\text{comp}}$, would approximately be fixed):

$$E_{\text{Dyn}} = (E_{\text{Kdisc}} + E_{\text{comp}}) \times E_{\text{nh}} \times \frac{\chi}{\tau}$$

where $\chi$ is the number of packets in a message, $\tau$ is the key refresh rate in packets per key, $E_{\text{Kdisc}}$ is the cost of shared key discovery with the next hop sensor after initial deployment, and $E_{\text{nh}}$ is the expected number of hops. In the dynamic key-based schemes, $\tau$ may change periodically, on demand, or after a node-compromise. Further improvements for dynamic key management and en-route filtering schemes without too much overhead are still possible by sharing a dynamically created cryptic credential. Specifically depending on a unique piece of information that sensor possesses, keys can be generated instead of being exchanged. For this, the residual battery life or energy on a node [7], virtual energies, local time in the node, or identity of the node could be utilized as the shared dynamic cryptic credential and would be updated approximately as necessary.

A good analytical lower bound for $E_{\text{nh}}$ is given in [9] as

$$E_{\text{nh}} = \frac{(D-tr)}{(E_{\text{dh}})} + 1$$

Where $D$ is the end-to-end distance (m) between the sink and the source sensor node, $tr$ is the approximated transmission range (m), and $E_{\text{dh}}$ is the expected hop distance.

2. Background and Motivation

Today, the wireless sensor networks are no longer nascent technology and future advances in technology will bring more sensor applications into our daily lives as well as into many diverse and challenging application scenarios. The main goal of this paper is to present general principles to aid in the design of secure WSN protocols. Suggestions for protocol designers to consider before an attempt to build the secure WSN protocols were first introduced in [8]. In this paper, we consider the following aspects of security services from the WSNs perspective

2.1. Confidentiality

Confidentiality refers to the protection of the exchanged content among the sinks and the sensors. An adversary having the privilege to access the content which should not be able to decode the exchanged messages in the network.

2.2. Authentication

In wireless sensor networks this mechanism verifies whether the exchanged information is emanating from the legitimate participant of WSN. This is needed because a malicious entity may be able to inject counterfeit content or resend the same content into the network. A classic technique to provide authentication would be to utilize Message Authentication codes (MACs).

2.3. Integrity

The recipients in the WSN should be able to detect if the exchanged content between the communicating participants of the WSN have been altered. It should ensure that the exchanged content is not deleted, replication of old data, counterfeit or stale. Integrity is provided by the usage of hashing algorithms.

2.4. Access Control

With access control, unauthorized use of a resource is prevented in WSNs. It addresses which participant of the network reaches which content or service.

The purpose of this paper is to develop an efficient and secure communication framework for WSN applications. Security (Encryption and keying) using Virtual Energy of the sensor node provides a technique
to verify data in line and drop false packets from malicious nodes, thus maintaining the health of the sensor network. This method dynamically updates keys without exchanging messages for key renewals and embeds integrity into packets as opposed to enlarging the packet by appending MACs. The contributions to this paper are:

- A dynamic en-route filtering mechanism that does not exchange explicit control messages for rekeying;
- Provision of one time keys for each packet transmitted to avoid stale keys;
- Modular and flexible security architecture with a simple technique for ensuring authenticity, integrity, and non-repudiation of data without enlarging packets with MACs;
- A robust secure communication framework that is operational in dire communication situations and over unreliable MACs. Both analytical and simulation results verify the feasibility.

3. Design

The overall mechanism is classified into three modules.

3.1 Keying Module

It is essentially the method used for handling the keying process. It produces a dynamic key that is then fed into the crypting module. In SVE, each sensor node has a certain virtual energy value when it is first deployed in the network. After deployment, sensor nodes traverse several functional states. The states mainly include node-stay-alive, packet reception, transmission, encoding, and decoding. As each of these actions occurs, the virtual energy in a sensor node is depleted. The current value of the virtual energy, \( E_{vc} \), in the node is used as the key to the key generation function, \( F \). During the initial deployment, each sensor node will have the same energy level \( E_{ini} \), therefore, the initial key, \( K_1 \), is a function of the initial virtual energy value and an initialization vector (IV). Subsequent keys, \( K_j \), are a function of the current virtual energy, \( E_{vc} \), and the previous key \( K_{j-1} \). SVEs virtual energy-based keying module ensures that each detected packet is associated with a new unique key generated based on the transient value of the virtual energy.

3.2 Crypting Module

Due to the resource constraints of WSNs, traditional digital signatures or encryption mechanisms requiring expensive cryptography is not viable. The encoding operation is essentially the process of permutation of the bits in the packet, according to the dynamically created permutation code via the RC4 encryption mechanism. The key to RC4 is created by the previous module (keying module). The purpose of the crypting module is to provide simple confidentiality of the packet header and payload while ensuring the authenticity and integrity of sensed data without incurring transmission overhead of traditional schemes. However, since the key generation and handling process is done in another module, SVEs flexible architecture allows for adoption of stronger encryption mechanisms in lieu of encoding.

3.3 Forwarding Module

The node after receiving the packet needs to follow the following steps:

- **Step1**: check for data received
- **Step2**: if yes Get Node Id
- **Step3**: if received node id = Check watched node
- **Step4**: send data to next node go to step 1
- **Step5**: decrypt data, check authenticity if authentic go to step 7
  - Else go to step 8
- **Step6**: Get current (my) Key value Encrypt data with My key value
- **Step7**: send data
- **Step 8**: go to step 1

The topology is taken with multiple clusters as shown in Fig1. All the sensor nodes communicate to their cluster heads which in turn send message to the sink node or the base station.

4. Operational Modes
According to the previous work, there are two modes of operations for watching a node in order to find the next node to where the packet is to be transmitted. In mode1 all nodes watch their neighbours; whenever a packet is received from a neighbour sensor node, it is decoded and its authenticity and integrity are verified. Only legitimate packets are forwarded towards the sink. When an event occurs and a report is generated, it is encoded as a function of a dynamic key based on the virtual energy of the originating node and transmitted. When the packet arrives at the next-hop node, the forwarding node extracts the key of the sending node (this could be the originating node or another forwarding node) from its record. (The virtual perceived energy value associated with the sending node and decodes the packet.) After the packet is decoded successfully, the plaintext ID is compared with the decoded ID. In mode2 nodes in the network are configured to only watch some of the nodes in the network. Each node randomly picks ‘r’ nodes to monitor and stores the corresponding state before deployment. As a packet leaves the source node (originating node or forwarding node) it passes through node(s) that watch it probabilistically. Thus, mode2 is a statistical filtering approach like SEF [2] and DEF [1].

5. Performance Evaluation

5.1 Attack Resilience

In this section, the performance of SVE is analyzed when there are malicious source nodes in the data collection field who insert bad packets into the network. Specifically, the analytical basis of the SVE framework’s resilience against malicious activities is formulated. Then, this theoretical basis is verified with the simulation results. We compare mode1 and mode2 considering the drop probability versus number of hops. We also take a closer look at VEBEK-II and how it is affected by the parameter, r (the number of records). In model1 and mode2, in order for an attacker to be able to successfully inject a false packet, an attacker must forge the packet encoding (which is a result of dynamically created permutation code via RC4). Given that the complexity of the packet is 2l, where l is the sum of the ID, TYPE, and DATA fields in the packet, the probability of an attacker correctly forging the packet is:

\[ P_{\text{forge}} = \frac{1}{2^{\text{packetsize}}} = \frac{1}{2^l} \]

Accordingly, the probability of the hacker incorrectly forging the packet, and therefore, the packet being dropped (P\text{drop-I}) is:

\[ P_{\text{drop-I}} = 1 - P_{\text{forge}} \]

Since mode1 authenticates at every hop, forged packets will always be dropped at the first hop with a probability of P\text{drop-I}. On the other hand, mode2 statistically drops packets along the route. Thus, the drop probability for mode2.

The attacker is identified and marked with a red circle and this is shown in fig2. , and the probability of dropping a packet from the node is shown in graph1.

![Fig. 1: Topology of WSN](image1)

![Fig. 2: Identifying the attacker and also dropping the malicious packets.](image2)

5.2 Energy Consumptions

In both the operational modes there is a single cost (E_{So}) to stay-alive, sense the event, encode the packet, and transmit the packet (E_{sa}, E_{sens}, E_{enc}, E_{tx}) at the source sensor. Thus,
5.3. Comparison with Other Statistical Schemes

We compare the expected energy costs of DEF, SEF and STEF with that of mode2 because mode2 is the statistical mode of SEV framework. The comparison results are illustrated in Graph3. As it can be seen from the above graphs that the mode2 has better performance than all the other schemes, and exhibiting a performance improvement of 60%-100% in energy consumption than the closest scheme, SEF.

6. Conclusion

Independent of the goal of saving energy, it may be very important to minimize the exchange of messages. It does not exchange control messages for key renewals and is therefore able to save more energy and is less chatty. It uses one key per message so successive packets of the stream use different keys making SEV more resilient to certain attacks (e.g., replay attacks, brute-force attacks, and masquerade attacks). It unbundles key generation from security services, providing a flexible modular architecture that allows for an easy adoption of different key-based encryption or hashing schemes. Compared the energy performance of our framework with other en-route malicious data filtering schemes.

7. References


